(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号

特開平7－105390

(43)公開日　平成７年(1995)４月21日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｔ　11/20					
Ｇ０９Ｇ　5/36	５２０　Ｐ	9471－5G			
		9192－5L	Ｇ０６Ｆ　15/ 72	３５５　Ｋ	

審査請求　未請求　請求項の数６　ＯＬ　（全 13 頁）

(21)出願番号　特願平5－244246

(22)出願日　平成５年(1993)９月30日

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中1015番地

(72)発明者　荒井　郁美
神奈川県横浜市港北区新横浜２丁目４番19号　株式会社富士通プログラム技研内

(72)発明者　馬場　匡史
神奈川県横浜市港北区新横浜２丁目４番19号　株式会社富士通プログラム技研内

(72)発明者　矢野口　晃大
神奈川県横浜市港北区新横浜２丁目４番19号　株式会社富士通プログラム技研内

(74)代理人　弁理士　穂坂　和雄　（外２名）

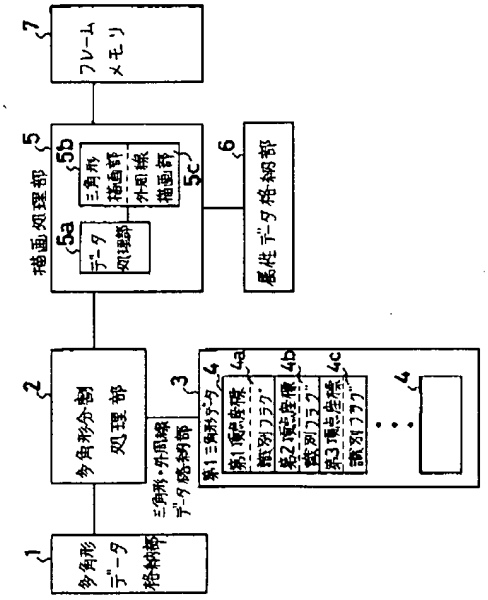
(54)【発明の名称】　外周線描画データ制御方式

(57)【要約】

【目的】本発明は多角形を複数の三角形に分割して描画を行う外周線描画データ制御方式に関し，多角形図形の外周線を描画する時に分割された三角形データを高速に処理し且つデータ量を少なくすること及び三角形をクリップ処理した時に分割された三角形の外周線データの処理を簡単化し，描画処理のために外周線データ送出する時に無駄なデータを削減し，データ送出時間を短縮化することを目的とする。

【構成】多角形データを入力して複数の三角形に分割する多角形分割処理部は，分割した各三角形データとして，それぞれの頂点座標と境界線か分割線の何れであるかを表す識別フラグが付加する。描画処理部は多角形分割処理部で作成して分割された各三角形データを受け取ると，識別フラグに基づいて各三角形の外周線を処理して多角形の描画を行うよう構成する。

本発明の第1の原理構成図



1

【特許請求の範囲】

【請求項1】　多角形を複数の三角形に分割して描画を行う外周線描画データ制御方式において，多角形データを入力して複数の三角形に分割する多角形分割処理部は，分割した各三角形のそれぞれの頂点座標と各頂点座標に対応する外周線が境界線か分割線の何れであるかを表す識別フラグとで構成する三角形データを作成し，描画処理部は前記多角形分割処理部で作成した前記分割された各三角形データを受け取って，前記識別フラグに基づいて各三角形の外周線を処理することにより多角形の描画を行うことを特徴とする外周線描画データ制御方式。

【請求項2】　請求項1において，前記描画処理部は，前記各三角形データの描画と前記境界線フラグが付加された頂点に対応する外周線だけを描画することを特徴とする外周線描画データ制御方式。

【請求項3】　請求項1において，前記外周線を描画する時，予め属性データ格納部に設定された描画条件を参照して外周線可視が設定されている外周線だけを描画することを特徴とする外周線描画データ制御方式。

【請求項4】　多角形を複数の三角形に分割してクリップ処理により切り出した上で描画を行う外周線描画データ制御方式において，多角形データを入力して複数の三角形に分割する多角形分割処理部は分割した各三角形データを作成し，クリップ処理を行うクリップ処理部は前記多角形分割処理部で作成した三角形データに対しクリップ枠により三角形分割を行い，クリップにより発生した三角形に対し三角形の各外周線を描画するか否かを表す外周線フラグを備えた三角形データを発生し，前記各外周線フラグを用いてクリップにより発生した三角形データを描画処理部に送出する時，前記各外周線フラグを用いて各三角形の頂点データを連結することを特徴とする外周線描画データ制御方式。

【請求項5】　請求項4において，前記クリップ処理部は，クリップ処理される前の元の三角形の外周線であるか否かにより前記外周線フラグを設定し，クリップにより分割された三角形の対応する各頂点データ毎に外周線フラグを付加し，前記三角形の各頂点毎の外周線フラグを加工して描画される外周線のパターンを表す複数ビットで構成するラインフラグを生成する外周線線形制御部を備えることを特徴とする外周線描画データ制御方式。

【請求項6】　請求項5において，前記クリップ処理部は，前記ラインフラグのパターンに基づいて，描画処理のために外周線分の頂点データの連結を行って図形データ格納部に格納する外周線連結部を備えることを特徴とする外周線描画データ制御方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は多角形を三角形に分割・描画する時の外周線描画データ制御方式に関する。

2

【0002】多角形を描画するために，多角形を複数の三角形に分割して各三角形の単位で描画を行う方式が利用されている。この場合に三角形を高速に処理して描画する時に，各三角形の各辺が元の多角形を形成する辺（境界線という）であるか，分割により形成された辺（分割線という）であるかを判別して，外周線の描画を行う必要があるため，独立して外周線の作成・描画が行われている。

【0003】また，多角形を分割して発生した三角形をクリップ処理（指定したサイズ及び位置の四角形の枠で図形を切り取る処理）で分割した三角形を発生してその三角形の外周線データを描画する場合があるが，効率良く外周線データを転送処理することが望まれている。

【0004】

【従来の技術】図12は従来の外周線描画の構成図，図13は従来例による外周線描画の動作例である。

【0005】図12において，多角形データ格納部100に，描画される多角形のデータが格納されている。この例では，図13のA．に示す5角形のデータが格納されており，多角形データ101として，第1頂点（図13のA．の1で示す）乃至第5頂点（図13のA．の5で示す）の各頂点の座標（X，Y，Z）が格納されている。多角形分割処理部102は，多角形データ101に対し，先ず三角形分割処理部103で5角形を3つの三角形に分割処理を行う。これにより図13のA．の5角形は，頂点1，2，3で構成する三角形，頂点1，3，4で構成する三角形及び頂点1，4，5で構成する三角形の3つに分割され，各三角形のデータは，三角形／外周線データ格納部105の中の三角形データ106として格納される。

【0006】一方，5角形の外周線（輪郭を表す線）は，外周線データ作成処理部104により作成され，作成されたデータは，三角形／外周線データ格納部105の中の外周線データ107として格納される。なお，この外周線データ107は第1頂点乃至第5頂点の座標により構成される。

【0007】三角形／外周線データ格納部105の三角形データ106は各三角形を単位に描画処理部108に転送され，外周線データ107も同様に転送される。次に描画処理部108の各部において図形の描画が行われる。まず，三角形データ処理部109において，三角形データの第1三角形について頂点データを解析し，三角形描画データを作成する処理を行い，続けて三角形描画部111でフレームメモリ114に第1三角形の描画を行う。図13のB．はフレームメモリへ第1三角形が描画された状態を示す。同様に，三角形データ処理部109と三角形描画部111により第2三角形，第3三角形の各三角形のデータ処理と描画が実行され，図13のC．及びDに各三角形が順次追加して描画された状態が示される。

【0008】一方，描画処理部108の外周線データ処理部110は，属性データ格納部113に格納された描画条件の『外周線可視／不可視』が何れに設定されるかによって処理を行うか否かが決定される。この描画条件の中には各種の条件が用意され，ユーザが描画を行う前に選択して設定することができ，外周線可視が設定されていると，5角形を描画する時，輪郭である外周線を明確に表示するよう描画し，不可視が設定されていると外周線を描画せず，また5角形の内部を網かけや，色等により描画できる。

【0009】外周線可視が設定されている場合，外周線データ処理部110は，外周線データ107の第1頂点と第2頂点の各データを解析し，外周線データを作成し，作成されたデータは外周線描画部112によりフレームメモリ114上に描画される。図13のE．にこの外周線が描画された状態を示す。次に第2頂点と第3頂点について同様の処理により描画が行われ，以下，第3頂点と第4頂点，第4頂点と第5頂点，第5頂点と第1頂点の各外周線について順次描画が行われる。図13のF．～I．にその時の各外周線が描画される様子を示す。

【0010】次に図形データから多角形分割した三角形データからクリップ処理で分割した三角形の外周線データを描画する方式を説明する。図14は従来のクリップ処理で分割した三角形の外周線データの制御方式の説明図である。図14の図形データ120は多角形を表すデータであり，このデータが多角形分割処理部121（図12の102と同様）に入力されると，多角形を複数の三角形に分割する処理が行われる。この分割処理により，多角形は複数の三角形に分割されて，各三角形の頂点データを含む三角形データ122が作成されると共に外周線データ123が作成される。これらのデータは，クリップ処理部124において，順番に送出された頂点データとしてクリップ枠との交点計算が行われ，新しく求めた頂点データを図形データ格納領域128に格納し，次いで描画処理部125に格納していた。

【0011】この場合のクリップ処理部124の機能を説明すると，図14の左下部に示すように，A，B，Cの3つの頂点を持つ三角形を，a～dの角を持つクリップ枠でクリップ（切り出し）を行い，クリップ枠と三角形の交点である座標e，fが求められると，このクリップで形成された三角形の新たな頂点データ（A，e，f）を図形データ格納領域128の三角形データ129に格納していた。

【0012】また，クリップにより分割された三角形の外周線データを描画処理部125に送出する場合，外周線データ123として，図14に示すように，外周線線分L1（頂点Aと交点e間の線分）のデータとして頂点S1（線分L1のスタート位置を表し頂点Aの座標）と頂点E1（線分L1の終点を表し交点eの座標）の2頂点データが送られ，更に外周線線分L2のデータとして頂点S2（スタート位置である交点fの座標）と頂点E2（終点である頂点Aの座標）の2頂点データを順番に描画処理部125に送出し，これにより外周線描画を保障していた。

【0013】

【発明が解決しようとする課題】上記図12，図13に示した従来の外周線描画方式によれば，描画処理部は，多角形分割処理部で任意個数に分割された三角形のそれぞれを，三角形データ処理部において1三角形の単位で各頂点データに従って処理するだけで，その三角形データが多角形の一部であるかまたは独立した三角形であるかを意識せずに処理している。

【0014】また，外周線については，図形描画部の外周線データ処理部において，描画属性（可視／不可視）の条件により「可視」の場合に描画されるが，多角形図形の外周線を描画する場合，従来では元の多角形を形成する辺（以後，境界線という）であるか，三角形に分割した際に形成された辺（以後，分割線という）であるかを判別しないため，外周線は三角形とは独立にデータ作成及び描画処理を行う必要があり，データを格納するメモリ容量が多くなり，処理に時間がかかるという問題があった。

【0015】また，上記図14に示す従来のクリップされた三角形の外周線データは，先に外周線線分L1の頂点データとして頂点S1，頂点E1を描画処理部に送出した後，外周線線分L2のデータとして頂点S2と本来頂点S1と同一である頂点E2のデータを別の外周線線分のデータとして新たに描画処理部に送出するため，データ送出に無駄が生じるという問題があった。

【0016】本発明は多角形図形の外周線を描画する場合に，分割された三角形データを高速に処理し且つデータ量を少なくすることができる外周線描画データ制御方式を提供することを第1の目的とし，クリップ処理部において三角形をクリップした時の外周線データの制御と，描画処理部に外周線データ送出する時に，無駄なデータを削減し，データ送出時間を短縮化できる外周線データ制御方式を提供することを第2の目的とする。

【0017】

【課題を解決するための手段】図1は本発明の第1の原理構成図であり，図2は本発明の第2の原理構成図である。

【0018】図1において，1は多角形の頂点データを含む多角形データ格納部，2は多角形を複数の三角形に分割する多角形分割処理部，3は三角形・外周線データ格納部，4は三角形・外周線データ格納部3に格納された第1三角形データ，第2三角形データ・・・等の三角形データ，4a～4cは各三角形に対し第1乃至第3の各頂点座標と各頂点に対応する辺が境界線（多角形の外周線）か分割線（多角形を分割した時に生じた線）かを

表示する識別フラグとで構成する各頂点データ，5は描画処理部，5aは三角形データを処理するデータ処理部，5bは三角形描画部，5cは外周線描画部，6は描画条件を含む属性データ格納部，7はフレームメモリである。

【0019】また図2において，10は多角形データ格納部，11は多角形分割処理部，12はクリップ処理部，13はクリップ処理部12に設けられた外周線線形制御部，14は同じくクリップ処理部12に設けられた外周線連結処理部，15は図形データ格納部，16は描画処理部，17はフレームメモリである。この図2の10，11，16及び17の各部は，それぞれ上記図1の1，2，5及び7に対応する。

【0020】本発明の第1の構成は，多角形を分割するとき，三角形の各頂点データに次の頂点と結ぶ辺が境界線であるか分割線であるかを示すフラグを付加し，描画処理部において高速に外周線を描画処理できるようにしたものである。また，第2の構成では分割された三角形に対しクリップ枠によるクリップ処理において三角形分割を行い，その三角形データの中に各頂点に対応する外周線を描画するか否かを表すフラグを付加し，そのフラグを用いて外周線描画のタイプを識別し，描画処理部に対し必要な外周線のデータだけを送出するものである。

【0021】

【作用】図1の構成において，多角形分割処理部2は多角形データ格納部1の多角形データを入力し，多角形を複数の三角形に分割する。この分割により作成された各三角形の頂点に対し境界線・分割線を示す識別フラグを付加し，三角形データ格納部3の三角形データ4として格納する。三角形データ4は3つの頂点データ4a～4cで構成され，頂点座標と識別フラグとで構成される。

【0022】三角形データ4は描画処理部5に転送され，この中のデータ処理部5aで各頂点データが解析され，三角形描画部5bでフレームメモリ7に三角形が描画され，外周線描画部5cで属性データ格納部6の描画条件（外周線可視・不可視）を参照し，可視の場合は境界線を表すフラグが付された頂点だけフレームメモリ7に外周線を描画する。描画処理部5は，全ての三角形について同様に描画を行う。

【0023】図2の構成において，多角形分割処理部11は上記図1と同様に多角形データ格納部10の多角形データを入力し複数の三角形に分割する。分割された各三角形データに対し，クリップ処理部12はクリップ枠でクリップ処理を行い，枠に入る図形を三角形分割して，分割された三角形の各頂点座標と各頂点に対応する外周線が可視か不可視かを表す外周線フラグが付加された三角形データが作成される。外周線線形制御部13には各頂点に対応する外周線フラグが入力され，外周線フラグを用いて，三角形の描画タイプ（3辺の中のどの辺を描画するかのパターン）を表すラインフラグを作成する。次に外周線連結処理部14が，前記ラインフラグに基づいて，描画すべき外周線線分についてだけ頂点データの連結を行って，図形データ格納部15に格納する。この図形データ格納部15の内容は描画処理部16に送られる。描画処理部16は三角形描画と外周線描画の処理を行いフレームメモリ17に格納する。

【0024】

【実施例】図3は実施例1の構成図，図4は三角形データの構成例，図5は実施例1による5角形の描画の例である。

【0025】この実施例1は上記図1に示す本発明の第1の原理構成に対応する構成である。図3において，2～4，5及び7は上記図1と同じ名称を備えた装置であり，特に描画処理部5の内部の詳細が示されている。

【0026】以下に実施例1の動作を説明すると，多角形分割処理部2は図示されない多角形データ格納部（従来例の図12の100と同じ）からの多角形データ（同図12の101）を受け取って，これを複数の三角形に分割する処理を行い，分割された各三角形のデータを三角形・外周線データ格納部3に格納する。この時，各三角形について第1三角形データ，第2三角形データ・・と順次作成される。例えば，多角形データとして5角形データが入力した場合，第1，第2，第3の三角形データが作成される。

【0027】本発明の多角形分割処理部2は，各三角形データとして，第1頂点座標乃至第3頂点座標を発生すると共に，各頂点座標に対しその頂点に対応する外周線が境界線か分割線かの何れであるかを表す識別フラグを設定する。この識別フラグは，各頂点に対応する外周線が，元の多角形の境界を表す場合は境界線フラグとして“1”が設定され，分割により生じた外周線であれば分割線フラグとして“0”が設定される。

【0028】図4に示す三角形データの構成例では，一つの頂点座標として，40～42にX，Y，Zの各座標データ（各32ビット）が設定され，これにフラグデータ43が設定される。フラグデータ43は，0ビットから31ビットの合計32ビットで構成され，その中の16ビットに“1”（境界線フラグ）か“0”（分割線フラグ）が設定される。

【0029】このように，多角形分割処理部2により作成された各三角形のデータ4a～4cは，三角形・外周線データ格納部3内に各頂点座標と識別フラグと共に図のように格納される。

【0030】三角形・外周線データ格納部3は1三角形を単位に三角形データを描画処理部5に転送する。描画処理部5では，三角形・外周線データ入力部50で三角形データを入力し，データ処理部51（図1の5aに対応）で処理が行われる。データ処理部51の処理のフローが図3のデータ処理部51内に示されているように，最初に第n三角形データ解析（nは最大で分割により作

成された三角形の数までの任意の数）において，第ｎ番目の三角形の各頂点データが解析され（図３の５１内のＳ１），解析結果に基づいて三角形・外周線描画データが作成される（同Ｓ２）。この場合，三角形の頂点座標と対応する外周線の識別フラグが一体となって作成される。作成されたデータは描画部５２に送られる一方，他の三角形データが有るか判別し（同Ｓ３），有る場合は，次の三角形・外周線データについて同様の処理が行われる。

【００３１】描画部５２は送られた三角形・外周線描画データについて，三角形描画部５３（図１の５ｂに対応）において第ｎ三角形描画を行う。最初は第１三角形をフレームメモリ７に描画し，続けて外周線描画部５４でその第１三角形の外周線の描画を行う。この時，属性データ格納部６に格納された描画条件（外周線可視／不可視）を参照し，外周線可視で且つ境界線フラグが付加された頂点に対応する外周線だけをフレームメモリ７に描画する。

【００３２】データ処理部５１から次の第２三角形の三角形・外周線描画データが送られてくると，同様にその三角形と外周線を描画し，全ての三角形・外周線描画データについてフレームメモリ７上に描画される。

【００３３】図５の例で説明すると，Ａ．に示す５角形を３つの三角形に分割して描画する場合，第１三角形は頂点１，２，３の各頂点座標と，頂点１に対応する外周線（頂点１と頂点２を結ぶ線）は境界線フラグが付加され，頂点２に対応する外周線（頂点２と頂点３を結ぶ線）も境界線フラグが付加され，頂点３に対応する外周線（頂点３と頂点１を結ぶ線）は分割線フラグが付加されている。

【００３４】このような第１三角形のデータが，データ処理部５１で処理されて描画部５２において描画した結果，図５のＢ．に示すように頂点１，２，３で構成する第１三角形（網かけ表示）と太線で示す２つの外周線（頂点１，２間及び頂点２，３間）が一緒に描画される。この後，一本の境界線フラグを持つ外周線（頂点３と４の間）を表す頂点３のデータを備えた第２三角形の描画が図５のＣ．に示すように描画され，最後に第３三角形が図５のＤ．のように描画されて，５角形全体の描画が完成する。

【００３５】次に図６乃至図１０は上記図２に示す本発明の第２の原理構成に対応する実施例２の構成に関する図である。図２に示す構成において，多角形分割処理部１１で分割された一つの三角形に対して，クリップ処理部１２で従来と同様にクリップ枠と三角形との交点を求めてクリップ処理が行われる。この具体例を図６により説明する。

【００３６】図６はクリップ処理によりクリップされる三角形の例であり，分割された三角形は頂点Ａ，Ｂ，Ｃで構成され，この三角形とクリップ枠Ｄとは点Ｂ’と点Ｃ’で交差する。従ってクリップ処理により，新たに頂点Ａ，Ｂ’，Ｃ’で構成する三角形が，元の頂点Ａ，Ｂ，Ｃで構成する三角形から分割される。

【００３７】クリップ処理部１２では，クリップにより分割された三角形（図６の頂点Ａ，Ｂ’，Ｃ’の三角形）の各外周線を描画（可視）か，描画無し（不可視）かを表す外周線フラグを作成する。すなわち，図６の線分Ｌ１（頂点Ａと頂点Ｂ’間の外周線）と線分Ｌ２（頂点Ａと頂点Ｃ’間の外周線）は，共に元の三角形の外周線上の線分であるから可視（描画）を表す状態に外周線フラグが設定されるが，頂点Ｂ’と頂点Ｃ’間の線分はクリップ処理による分割で生じた外周線であるから描画しない状態に外周線フラグを設定される。

【００３８】図７は外周線フラグの説明図であり，図７のＡ．は外周線フラグの構成図，Ｂ．は頂点と外周線フラグの関係を示す。Ａ．に示すように外周線フラグは，３２ビットのデータ中の０ビットの位置に設定され，この０ビットが“１”の時外周線可視（描画有り），“０”の時外周線不可視（描画無し）を表す。この外周線フラグは，Ｂ．に示すように頂点ｎと頂点ｎ＋１の外周線（ｅｎ：エッジｎを意味する）の外周線フラグが頂点ｎの座標データに付加され，同様に頂点ｎ＋１の座標データにｅｎ＋１（エッジｎ＋１の外周線フラグが付加され，頂点ｎ＋２にｅｎ＋２（エッジｎ＋２）の外周線フラグが付加される。

【００３９】図８及び図９は実施例２の処理フロー図（その１），（その２）である。この処理フローはクリップ処理部（図２の１２）の制御により動作する外周線線形制御部（図２の１３）及び外周線連結処理部（図２の１４）の機能を示す。

【００４０】また，この処理フローは，クリップ処理部により図８の右側に示す頂点Ｐ１，Ｐ２，Ｐ３を備えた三角形が作成され，各エッジの外周線フラグが上記，図７のＢ．と同様の関係で各頂点の座標データに付加されているものとし，各エッジの外周線フラグは“０”または“１”の何れか任意に設定されている。

【００４１】図８の処理が開始すると，外周線線形制御部の処理が実行され，内部のメモリ（図示せず）の特定エリアに設けられたラインフラグ（LINEFLG と表示）という名前の変数（メモリの特定のエリア）を０に初期化する（図８のＳ１）。次に，ステップＳ２において，以下の処理が実行される。すなわち，最初にラインフラグ（LINEFLG）に対し，エッジ１（edge1)の外周線フラグの内容を設定する。この時，上記図７のＡ．に示すように，３２ビット内の０ビット位置に設定されたフラグを表す内容（１または０）がラインフラグ（LINEFLG）の０ビットに設定する。

【００４２】次に，エッジ２（edge2)の外周線フラグを左側に１ビットシフトして，その外周線フラグのビット（左に１ビットシフトしているので，３２ビット内の１

ビット位置にある）をラインフラグ（LINEFLG）の対応する1ビット位置に設定する。さらに，エッジ3（edge 3)の外周線フラグを左に2ビットシフトし，そのラインフラグのビット（32ビット内の2ビット位置にある）をラインフラグ（LINEFLG）に設定する。

【0043】図10に示すラインフラグの説明図により，上記ステップS2の処理内容を説明する。図10のA．において，①～③はエッジ1～エッジ3の外周線フラグが対応する左シフトを行った後の状態を示し，④はラインフラグを表す。①～③のそれぞれの0ビット位置（フラグビット）の内容が④に示すようにラインフラグの3つのビット位置に設定される。

【0044】図8に戻って，次に得られたラインフラグ（LINEFLG）について，内容が0か否か判断する（図8のS3）。この判断でラインフラグ（LINEFLG）の内容が全て0（全ビットが“0”）の場合は，描画する外周線が無いので外周線連結処理をすることなく終了する。内容が0でない場合，ラインフラグ（LINEFLG）によりジャンプ先を設定し（図8のS4），設定されたジャンプ先（M1～M7の何れか：M1～M7については後述する）にジャンプする（同S5）。このジャンプ先は，図9に示す外周線連結処理部である。

【0045】ここで，上記ラインフラグ（LINEFLG）の作用を図10により説明すると，図10のA．によりラインフラグ（LINEFLG）の3ビットには，最下位の0ビット位置にエッジ1の外周線フラグ，1ビット位置にエッジ2の外周線フラグ，2ビット位置にエッジ3の外周線フラグがそれぞれ設定されている。このラインフラグ（LINEFLG）の3ビットで表す範囲は000～111であり，10進数の0～7の8つの値になる。

【0046】図10のB．の表はこのラインフラグ（LINEFLG）の8つの値が，それぞれ外周線描画線分の描画のパターンを表すことを示し，これに対応する外周線描画線分のパターン例を図11に示す。

【0047】すなわち，図10のB．において，番号(1) は，ラインフラグが0の場合であり，描画無しであって，ジャンプ先はM0（図8のステップS3で0と判断された場合）となり，外周線描画線分は図11の(1) に示すように転送されない。また，番号(2) は，ラインフラグが1の場合であり，外周線描画線分は図8の右側に示す三角形の頂点P1－P2間を描画することを指示し，使用する外周線エッジはエッジ1であり，ジャンプ先はM1となる。この場合，ジャンプ先で図11の(2) に示す描画が行われる。

【0048】同様に，番号(3) ～(8) はラインフラグ（LINEFLG）の「2」～「7」の値に対応して，図10のB．の各外周線描画線分に示す各頂点区間が，対応する外周線エッジを使用して描画することを指示し，それぞれのジャンプ先はラインフラグ（LINEFLG）の数値に対応するM2～M7である。

【0049】図9には上記図8の外周線線形制御部からジャンプされる外周線連結処理部の処理フローが示され，ジャンプ先として設定されたM1～M7の何れかに応じて対応する外周線連結処理が実行されて，実行後図8に戻り終了する。なお，M1～M7は上記図10のB．の各番号(2) ～(7) の場合に該当する。

【0050】例えば，ジャンプ先がM1の場合，第1頂点（図8の三角形のP1）のデータを図形データ格納部（図2の15）に格納し，次に第2頂点（図8の三角形のP2）のデータを格納する。これにより，図11の(2) に示すような外周線が描画される。

【0051】また，上記図6に示すクリップにより作成された三角形の2つの外周線を描画する場合は，図8の三角形の頂点P2－P3－P1間を描画する場合に相当する。この時，ラインフラグ（LINEFLG）の3ビットは“110”に設定され，10進数の「6」であるからM6にジャンプする。この場合，図9のフローにより，第2頂点（P2）のデータの格納（図形データ格納部へ），第3頂点のデータの格納，第1頂点のデータの格納が順次行われ，3つの頂点データが格納されると，これらのデータが描画処理部（図2の16）でフレームメモリに描画される。

【0052】この場合，描画処理部へ転送される頂点データは2つの外周線線分が連結されて3個であり，従来例のように各線分毎に始点－終点の2つの頂点データを転送する必要がなく，メモリ容量を少なくし，転送時間を短縮することができる。

【0053】

【発明の効果】本発明の第1の原理によれば，多角形を複数の三角形に分割する際に境界線・分割線を表す識別フラグを頂点データに付加して，これを処理することにより描画できるので，外周線処理を高速化すると共にメモリを縮小化することができる。すなわち，外周線データの作成処理を簡単化し，多角形分割処理部と描画処理部間でのデータ転送量を削減し，描画処理における外周線データ処理を削減することができる。

【0054】本発明の第2の原理によれば三角形の外周線を描画するとき，多角形分割した三角形がクリップされた後，複雑な三角形の外周線頂点データを管理する場合に外周線フラグを用いてデータを簡略化することが可能となる。また，描画に必要な外周線頂点データを削減し，データ送出時間を短縮することができる。

【図面の簡単な説明】

【図1】本発明の第1の原理構成図である。

【図2】本発明の第2の原理構成図である。

【図3】実施例1の構成図である。

【図4】三角形データの構成例を示す図である。

【図5】実施例1による5角形の描画の例を示す図である。

【図6】クリップ処理によりクリップされる三角形の例

を示す図である。

【図７】外周線フラグの説明図である。

【図８】実施例２の処理フロー図（その１）である。

【図９】実施例２の処理フロー図（その２）である。

【図１０】ラインフラグの説明図である。

【図１１】外周線描画線分のパターン例を示す図である。

【図１２】従来の外周線描画の構成図である。

【図１３】従来例による外周線描画の動作例を示す図である。

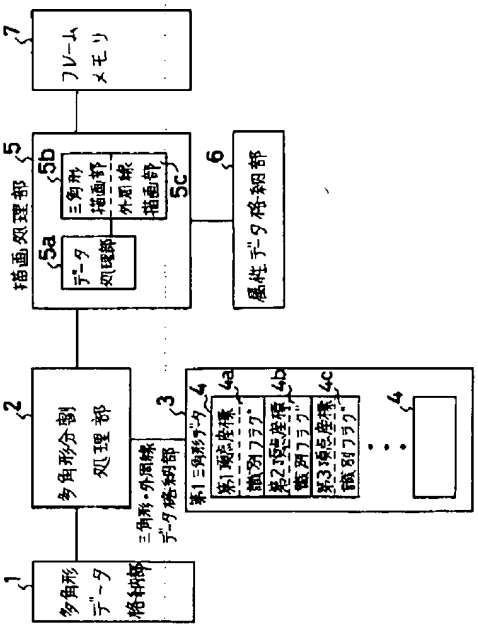
【図１４】従来のクリップ処理で分割した三角形の外周線データの制御方式の説明図である。

【符号の説明】

1	多角形データ格納部
2	多角形分割処理部
3	三角形・外周線データ格納部
4	三角形データ
4a～4c	各頂点データ
5	描画処理部
5a	データ処理部
5b	三角形描画部
5c	外周線描画部
6	属性データ格納部
7	フレームメモリ

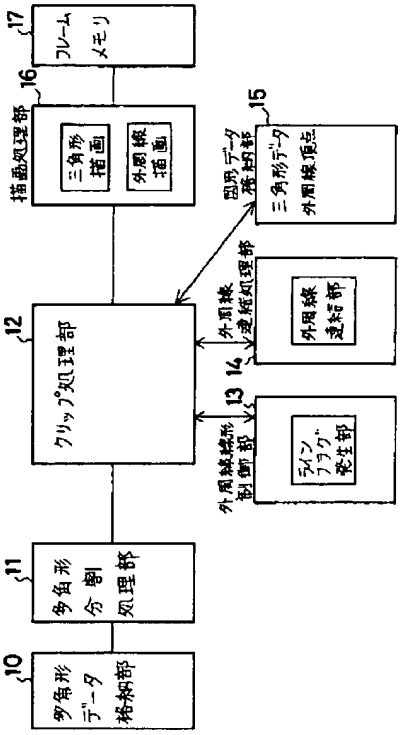
【図１】

本発明の第1の原理構成図



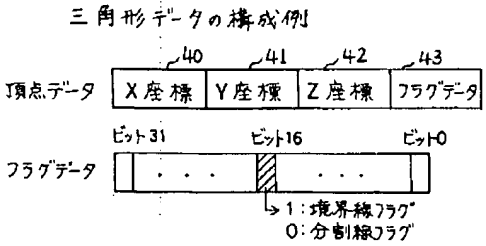
【図２】

本発明の第２の原理構成図



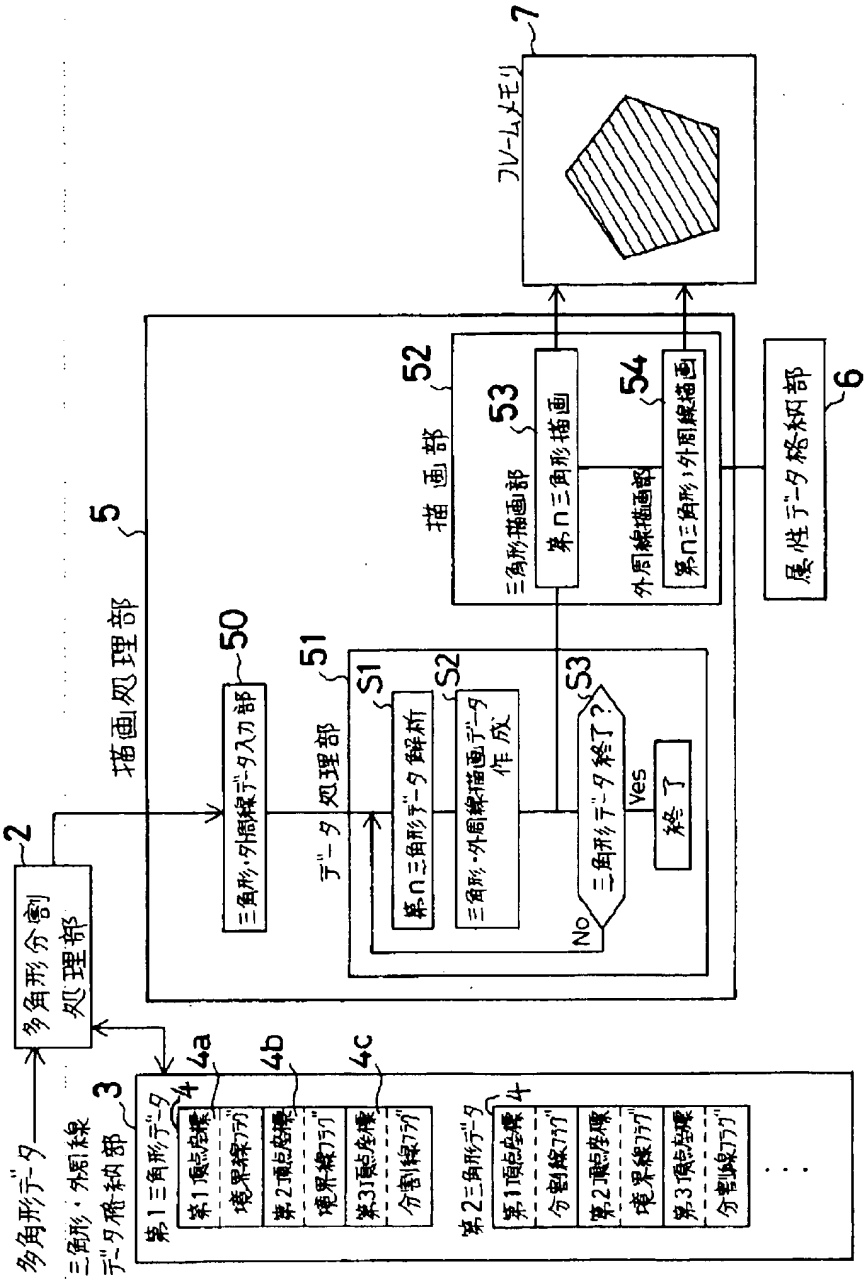
【図４】

三角形データの構成例



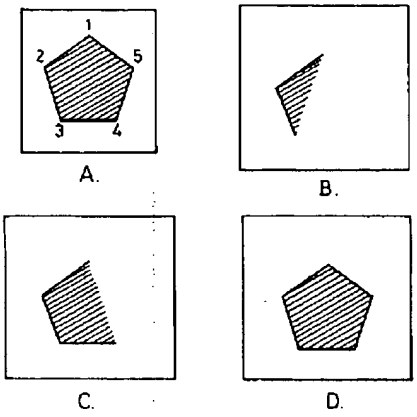
【図3】

実施例1の構成図


多角形データ
三角形・外周線データ格納部
3
第1三角形データ
4
4a
第1頂点座標
境界線フラグ
4b
第2頂点座標
境界線フラグ
4c
第3頂点座標
分割線フラグ
第2三角形データ
4
第1頂点座標
分割線フラグ
第2頂点座標
境界線フラグ
第3頂点座標
分割線フラグ
多角形分割処理部
2
描画処理部
5
三角形・外周線データ入力部
50
データ処理部
51
S1
第n三角形データ解析
S2
三角形・外周線描画データ作成
S3
三角形データ終了?
No
Yes
終了
描画部
52
三角形描画部
53
第n三角形描画
外周線描画部
54
第n三角形の外周線描画
フレームメモリ
7
属性データ格納部
6

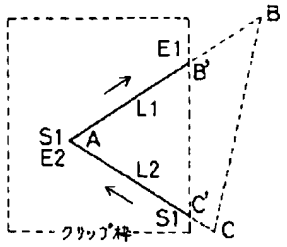
【図5】

実施例1による5角形の描画の例



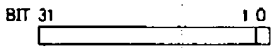
【図6】

クリップ処理によりクリップされる三角形の例



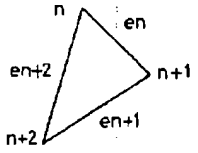
【図7】

外周線フラグの説明図



0ビット　1：外周線可視(描画有り)
　　0：外周線不可視(描画無し)

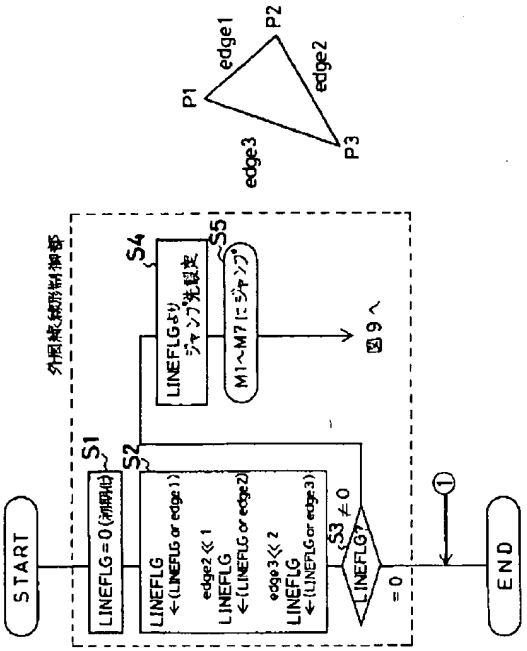
A. 外周線フラグの構成図



B. 頂点と外周線フラグの関係図

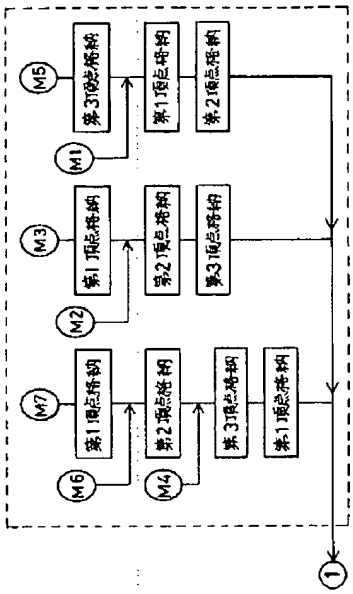
【図8】

実施例2の処理フロー図(その1)



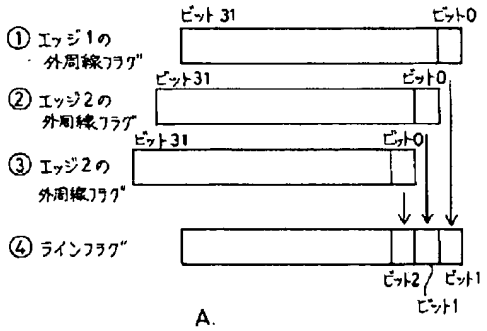
【図9】

実施例2の処理フロー図(その2)



【図10】

ラインフラグの説明図



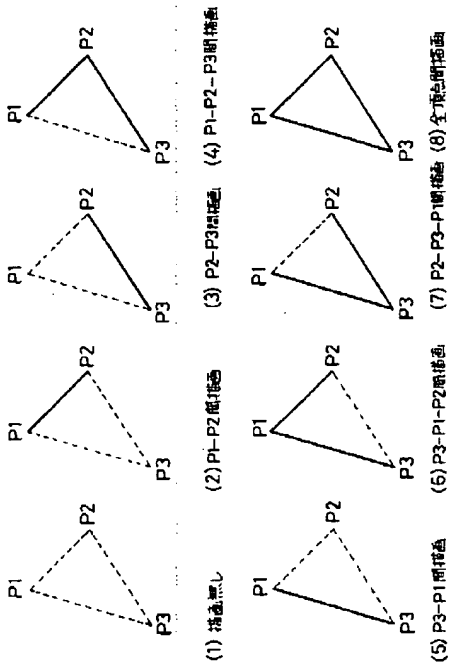
A.

	外周線描画,線分	使用する外周線エッジ	LINEFLG	ジャンプ先
(1)	描画無し	外周線エッジなし	0	M0
(2)	P1-P2間描画	edge1	1	M1
(3)	P2-P3間描画	edge2	2	M2
(4)	P1-P2-P3間描画	edge1,edge2	3	M3
(5)	P3-P1間描画	edge3	4	M4
(6)	P3-P1-P2間描画	edge3,edge1	5	M5
(7)	P2-P3-P1間描画	edge2,edge3	6	M6
(8)	全頂点間描画	edge1,edge2,edge3	7	M7

B.

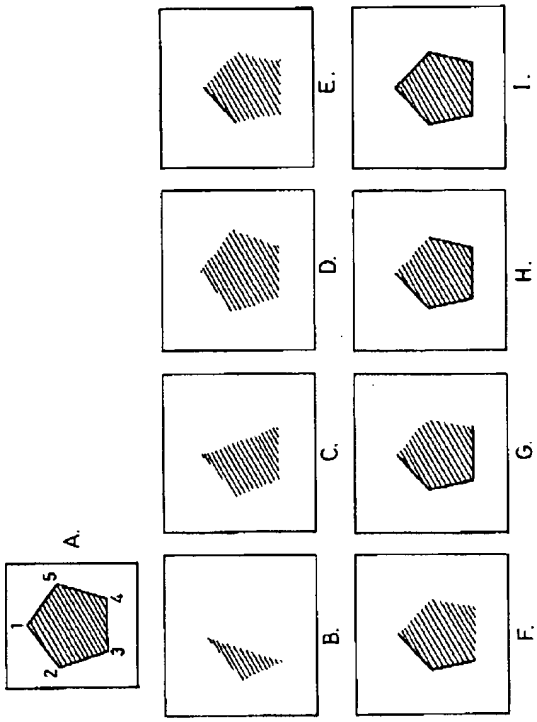
【図11】

外周線描画線分のパターン例を示す図



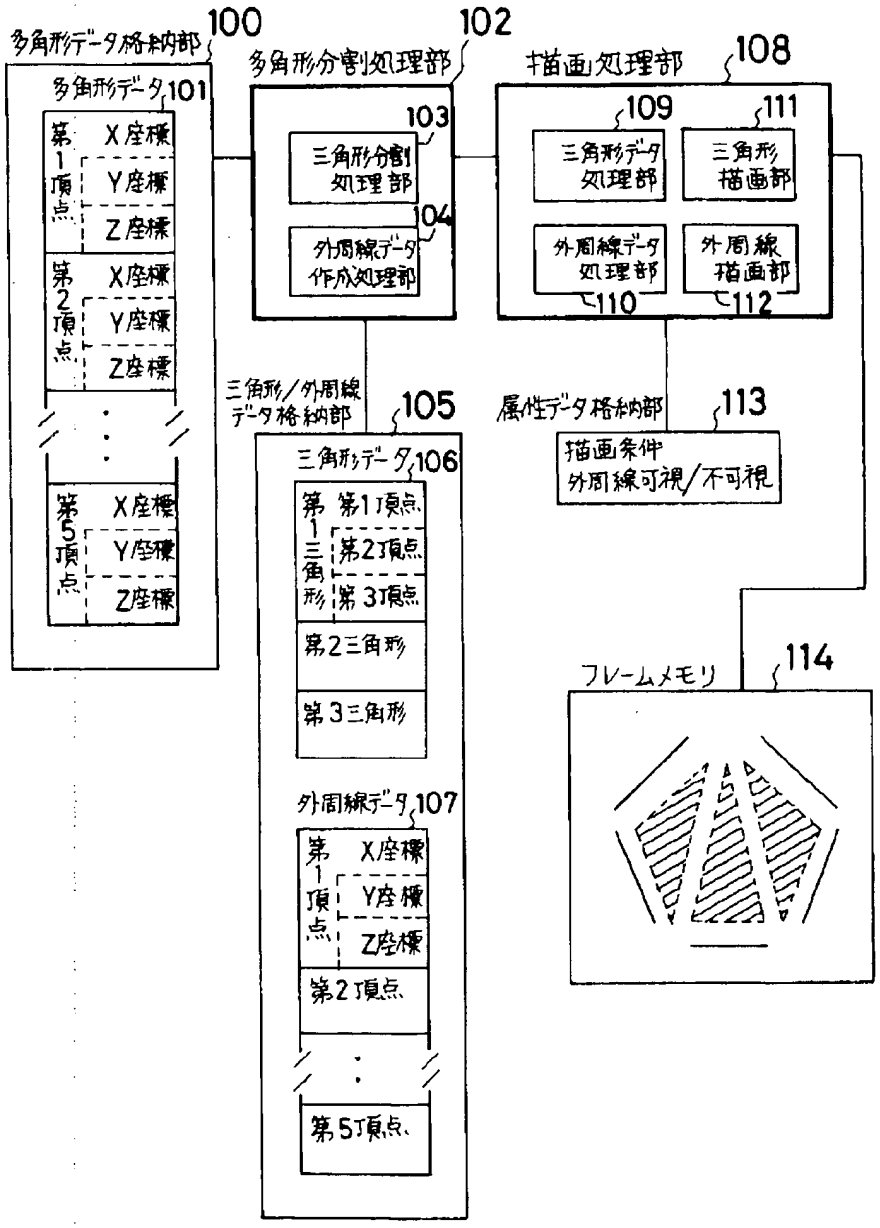
【図13】

従来例による外周線描画の動作例を示す図



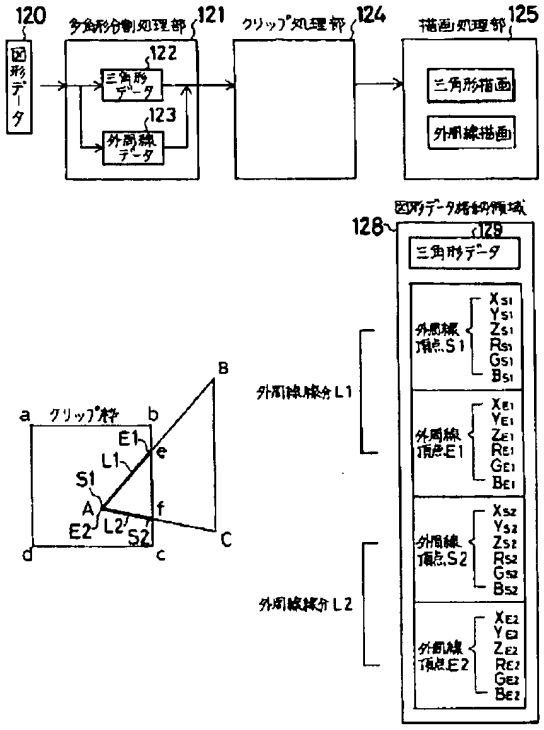
【図12】

従来の外周線描画の構成図



【図14】

従来のクリップ処理で分割した三角形の外周線データの制御方式の説明図



* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] According to the conventional periphery drawing drawing method shown in above-mentioned drawing 12 and drawing 13 , the drawing processing section only processes each of the triangle divided into the arbitration number in the polygon division processing section according to each top-most-vertices data in the triangle data-processing section in the unit of one triangle, and is processing it, without being conscious of whether the triangle data is polygonal [a part of] or it is the independent triangle.
[0014] Moreover, in a "visible" case, about a periphery line, are drawn by the conditions of a drawing attribute (visible/invisibility) in the periphery line data-processing section of the graphic form drawing section, but When drawing the periphery line of a polygon graphic form, in order not to distinguish whether it is the side (it is henceforth called a boundary line) which forms the original polygon in the former, or it is the side (it is henceforth called a parting line) formed when it divided into a triangle, The periphery line needed to perform data origination and drawing processing independently of the triangle, and its memory space which stores data increased, and it had the problem that processing took time amount.
[0015] Moreover, the periphery line data of the triangle which the former shown in above-mentioned drawing 14 clipped After sending out top-most vertices S1 and top-most vertices E1 to the drawing processing section as top-most-vertices data of the periphery line segment L1 previously, as data of the periphery line segment L2 Since the data of top-most vertices S2 and the top-most vertices E2 same originally as top-most vertices S1 were newly sent out to the drawing processing section as data of another periphery line segment, the problem that futility arose was in data forwarding.
[0016] This invention sets it as the 1st purpose to offer the periphery drawing drawing data control method which can process the divided triangle data at a high speed, and can lessen the amount of data, when drawing the periphery line of a polygon graphic form, and sets it in the clip processing section. When carrying out periphery line data forwarding to the drawing processing section with control of the periphery line data when clipping a triangle, useless data are reduced and it sets it as the 2nd purpose to offer the periphery line data control method which can shorten data forwarding time amount.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

EFFECT OF THE INVENTION

[Effect of the Invention] Since according to the 1st principle of this invention it can draw by adding the discernment flag showing a boundary line and a parting line to top-most-vertices data, and processing this in case a polygon is divided into two or more triangles, memory can be contraction-ized while accelerating periphery line processing. That is, creation processing of periphery line data can be simplified, the amount of data transfer between the polygon division processing section and the drawing processing section can be reduced, and periphery line data processing in drawing processing can be reduced.
[0054] When drawing a triangular periphery line according to the 2nd principle of this invention, after the triangle which carried out polygon division clips, when managing the periphery line top-most-vertices data of a complicated triangle, it becomes possible to use a periphery line flag and to simplify data. Moreover, periphery line top-most-vertices data required for drawing can be reduced, and data forwarding time amount can be shortened.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] The block diagram of the periphery drawing drawing of the former [drawing 12] and drawing 13 are the examples of the periphery drawing drawing by the conventional example of operation.
[0005] In drawing 12 , the data of the polygon drawn by the polygon data storage section 100 are stored. In this example, the data of five square shapes shown in A. of drawing 13 are stored, and the coordinate (X, Y, Z) of each top-most vertices of the 1st top-most vertices (1 of A. of drawing 13 shows) thru/or the 5th top-most vertices (5 of A. of drawing 13 shows) is stored as polygon data 101. The polygon division processing section 102 performs division processing for five square shapes to three triangles in the triangulation processing section 103 first to the polygon data 101. Thereby, five square shapes of A. of drawing 13 are divided into three of the triangles constituted from a triangle constituted from top-most vertices 1, 2, and 3, a triangle constituted from top-most vertices 1, 3, and 4, and top-most vertices 1, 4, and 5, and the data of three square shapes each are stored as triangle data 106 in a triangle / periphery line data storage section 105.
[0006] On the other hand, the periphery line (line showing a profile) of five square shapes is created by the periphery line data origination processing section 104, and the created data are stored as periphery line data 107 in a triangle / periphery line data storage section 105. In addition, this periphery line data 107 is constituted by the coordinate of the 1st top-most vertices thru/or the 5th top-most vertices.
[0007] Three square shapes each are transmitted to the triangle data 106 of a triangle / periphery line data storage section 105 by the unit at the drawing processing section 108, and the periphery line data 107 are transmitted similarly. Next, drawing of a graphic form is performed in each part of the drawing processing section 108. First, in the triangle data-processing section 109, top-most-vertices data are analyzed about the 1st triangle of triangle data, processing which creates triangle drawing data is performed and the 1st triangle is continuously drawn to a frame memory 114 in the triangle drawing section 111. B. of drawing 13 shows the condition that the 1st triangle was drawn, to a frame memory. Similarly, data processing of three square shapes each of the 2nd triangle and the 3rd triangle and drawing are performed by the triangle data-processing section 109 and the triangle drawing section 111, and the condition that three square shapes each carried out the sequential addition, and were drawn by C. of drawing 13 and D is shown.
[0008] On the other hand, it is determined whether "periphery line visible / invisibility" of drawing conditions by which the periphery line data-processing section 110 of the drawing processing section 108 was stored in the attribute data storage section 113 process by as any it is set. If it draws so that the periphery line which is a profile may be displayed clearly, when drawing five square shapes, if it can choose and set up and periphery line visible ****** is carried out and invisibility is set up before various kinds of conditions are prepared into this drawing condition and a user draws, a periphery line is not drawn and the interior of five square shapes can be drawn by shading, a color, etc.
[0009] When periphery line visible ****** is carried out, the periphery line data-processing section 110 analyzes each data of the 1st top-most vertices of the periphery line data 107, and the 2nd top-most vertices, and creates periphery line data, and the created data are drawn by the periphery line drawing

section 112 on a frame memory 114. The condition that this periphery line was drawn by E. of drawing 13 is shown. Next, drawing is performed by the processing same about the 2nd top-most vertices and the 3rd top-most vertices, and sequential drawing is hereafter performed about each periphery line of the 3rd top-most vertices, the 4th top-most vertices and the 4th top-most vertices, the 5th top-most vertices and the 5th top-most vertices, and the 1st top-most vertices. Signs that each periphery line at that time is drawn by F. of drawing 13 - I. are shown.

[0010] Next, the method which draws the triangular periphery line data divided by clip processing is explained from the triangle data which carried out polygon division from graphic data. Drawing 14 is the explanatory view of the control system of the triangular periphery line data divided by the conventional clip processing. The graphic data 120 of drawing 14 are data showing a polygon, and if this data is inputted into the polygon division processing section 121 (drawing 12 is the same as that of 102), processing which divides a polygon into two or more triangles will be performed. A polygon is divided into two or more triangles, and while the triangle data 122 containing the top-most-vertices data of three square shapes each are created by this division processing, the periphery line data 123 are created by it. In the clip processing section 124, intersection count with a clip frame was performed as top-most-vertices data sent out in order, and these data stored in the graphic form data storage field 128 the top-most-vertices data for which it asked newly, and, subsequently to the drawing processing section 125, stored them.

[0011] If the function of the clip processing section 124 in this case is explained, as shown in the left lower quadrant of drawing 14 If it clips by the clip frame with the angle of a-d (logging) and the coordinates e and f which are the intersections of a clip frame and a triangle are searched for, a triangle with three top-most vertices, A, B, and C Top-most-vertices data with the new triangle formed with this clip (A, e, f) were stored in the triangle data 129 of the graphic form data storage field 128.

[0012] Moreover, as periphery line data 123, when it sends out the triangular periphery line data divided with a clip to the drawing processing section 125, as shown in drawing 14 As data of the periphery line segment L1 (segment between top-most vertices A and Intersection e) Two top-most-vertices data of top-most vertices S1 (the start location of a segment L1 is expressed and it is the coordinate of top-most vertices A) and top-most vertices E1 (the terminal point of a segment L1 is expressed and it is the coordinate of Intersection e) are sent. Further as data of the periphery line segment L2 Two top-most-vertices data of top-most vertices S2 (coordinate of the intersection f which is a start location), and top-most vertices E2 (coordinate of the top-most vertices A which are terminal points) were sent out to the drawing processing section 125 in order, and this secured periphery drawing drawing.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Industrial Application] This invention relates to the periphery drawing drawing data control method when dividing and drawing a polygon at a triangle.
[0002] In order to draw a polygon, the method which divides a polygon into two or more triangles, and draws in the unit of three square shapes each is used. In this case, since each side of three square shapes each needs to distinguish whether it is the side (it is called a boundary line) which forms the original polygon, or it is the side (it is called a parting line) formed of division and needs to draw a periphery line when processing a triangle at a high speed and drawing, creation and drawing of a periphery line are performed independently.
[0003] Moreover, although the triangle which divided the triangle which divided the polygon and was generated by clip processing (processing which cuts off a graphic form by the frame of the square of the size and the location which were specified) is generated and the periphery line data of the triangle may be drawn, to carry out transfer processing of the periphery line data efficiently is desired.
[0004]
[Description of the Prior Art] The block diagram of the periphery drawing drawing of the former [drawing 12] and drawing 13 are the examples of the periphery drawing drawing by the conventional example of operation.
[0005] In drawing 12 , the data of the polygon drawn by the polygon data storage section 100 are stored. In this example, the data of five square shapes shown in A. of drawing 13 are stored, and the coordinate (X, Y, Z) of each top-most vertices of the 1st top-most vertices (1 of A. of drawing 13 shows) thru/or the 5th top-most vertices (5 of A. of drawing 13 shows) is stored as polygon data 101. The polygon division processing section 102 performs division processing for five square shapes to three triangles in the triangulation processing section 103 first to the polygon data 101. Thereby, five square shapes of A. of drawing 13 are divided into three of the triangles constituted from a triangle constituted from top-most vertices 1, 2, and 3, a triangle constituted from top-most vertices 1, 3, and 4, and top-most vertices 1, 4, and 5, and the data of three square shapes each are stored as triangle data 106 in a triangle / periphery line data storage section 105.
[0006] On the other hand, the periphery line (line showing a profile) of five square shapes is created by the periphery line data origination processing section 104, and the created data are stored as periphery line data 107 in a triangle / periphery line data storage section 105. In addition, this periphery line data 107 is constituted by the coordinate of the 1st top-most vertices thru/or the 5th top-most vertices.
[0007] Three square shapes each are transmitted to the triangle data 106 of a triangle / periphery line data storage section 105 by the unit at the drawing processing section 108, and the periphery line data 107 are transmitted similarly. Next, drawing of a graphic form is performed in each part of the drawing processing section 108. First, in the triangle data-processing section 109, top-most-vertices data are analyzed about the 1st triangle of triangle data, processing which creates triangle drawing data is performed and the 1st triangle is continuously drawn to a frame memory 114 in the triangle drawing

section 111. B. of drawing 13 shows the condition that the 1st triangle was drawn, to a frame memory. Similarly, data processing of three square shapes each of the 2nd triangle and the 3rd triangle and drawing are performed by the triangle data-processing section 109 and the triangle drawing section 111, and the condition that three square shapes each carried out the sequential addition, and were drawn by C. of drawing 13 and D is shown.

[0008] On the other hand, it is determined whether "periphery line visible / invisibility" of drawing conditions by which the periphery line data-processing section 110 of the drawing processing section 108 was stored in the attribute data storage section 113 process by as any it is set. If it draws so that the periphery line which is a profile may be displayed clearly, when drawing five square shapes, if it can choose and set up and periphery line visible ****** is carried out and invisibility is set up before various kinds of conditions are prepared into this drawing condition and a user draws, a periphery line is not drawn and the interior of five square shapes can be drawn by shading, a color, etc.

[0009] When periphery line visible ****** is carried out, the periphery line data-processing section 110 analyzes each data of the 1st top-most vertices of the periphery line data 107, and the 2nd top-most vertices, and creates periphery line data, and the created data are drawn by the periphery line drawing section 112 on a frame memory 114. The condition that this periphery line was drawn by E. of drawing 13 is shown. Next, drawing is performed by the processing same about the 2nd top-most vertices and the 3rd top-most vertices, and sequential drawing is hereafter performed about each periphery line of the 3rd top-most vertices, the 4th top-most vertices and the 4th top-most vertices, the 5th top-most vertices and the 5th top-most vertices, and the 1st top-most vertices. Signs that each periphery line at that time is drawn by F. of drawing 13 - I. are shown.

[0010] Next, the method which draws the triangular periphery line data divided by clip processing is explained from the triangle data which carried out polygon division from graphic data. Drawing 14 is the explanatory view of the control system of the triangular periphery line data divided by the conventional clip processing. The graphic data 120 of drawing 14 are data showing a polygon, and if this data is inputted into the polygon division processing section 121 (drawing 12 is the same as that of 102), processing which divides a polygon into two or more triangles will be performed. A polygon is divided into two or more triangles, and while the triangle data 122 containing the top-most-vertices data of three square shapes each are created by this division processing, the periphery line data 123 are created by it. In the clip processing section 124, intersection count with a clip frame was performed as top-most-vertices data sent out in order, and these data stored in the graphic form data storage field 128 the top-most-vertices data for which it asked newly, and, subsequently to the drawing processing section 125, stored them.

[0011] If the function of the clip processing section 124 in this case is explained, as shown in the left lower quadrant of drawing 14 If it clips by the clip frame with the angle of a-d (logging) and the coordinates e and f which are the intersections of a clip frame and a triangle are searched for, a triangle with three top-most vertices, A, B, and C Top-most-vertices data with the new triangle formed with this clip (A, e, f) were stored in the triangle data 129 of the graphic form data storage field 128.

[0012] Moreover, as periphery line data 123, when it sends out the triangular periphery line data divided with a clip to the drawing processing section 125, as shown in drawing 14 As data of the periphery line segment L1 (segment between top-most vertices A and Intersection e) Two top-most-vertices data of top-most vertices S1 (the start location of a segment L1 is expressed and it is the coordinate of top-most vertices A) and top-most vertices E1 (the terminal point of a segment L1 is expressed and it is the coordinate of Intersection e) are sent. Further as data of the periphery line segment L2 Two top-most-vertices data of top-most vertices S2 (coordinate of the intersection f which is a start location), and top-most vertices E2 (coordinate of the top-most vertices A which are terminal points) were sent out to the drawing processing section 125 in order, and this secured periphery drawing drawing.

[0013]

[Problem(s) to be Solved by the Invention] According to the conventional periphery drawing drawing method shown in above-mentioned drawing 12 and drawing 13 , the drawing processing section only processes each of the triangle divided into the arbitration number in the polygon division processing

section according to each top-most-vertices data in the triangle data-processing section in the unit of one triangle, and is processing it, without being conscious of whether the triangle data is polygonal [a part of] or it is the independent triangle.

[0014] Moreover, in a "visible" case, about a periphery line, are drawn by the conditions of a drawing attribute (visible/invisibility) in the periphery line data-processing section of the graphic form drawing section, but When drawing the periphery line of a polygon graphic form, in order not to distinguish whether it is the side (it is henceforth called a boundary line) which forms the original polygon in the former, or it is the side (it is henceforth called a parting line) formed when it divided into a triangle, The periphery line needed to perform data origination and drawing processing independently of the triangle, and its memory space which stores data increased, and it had the problem that processing took time amount.

[0015] Moreover, the periphery line data of the triangle which the former shown in above-mentioned drawing 14 clipped After sending out top-most vertices S1 and top-most vertices E1 to the drawing processing section as top-most-vertices data of the periphery line segment L1 previously, as data of the periphery line segment L2 Since the data of top-most vertices S2 and the top-most vertices E2 same originally as top-most vertices S1 were newly sent out to the drawing processing section as data of another periphery line segment, the problem that futility arose was in data forwarding.

[0016] This invention sets it as the 1st purpose to offer the periphery drawing drawing data control method which can process the divided triangle data at a high speed, and can lessen the amount of data, when drawing the periphery line of a polygon graphic form, and sets it in the clip processing section. When carrying out periphery line data forwarding to the drawing processing section with control of the periphery line data when clipping a triangle, useless data are reduced and it sets it as the 2nd purpose to offer the periphery line data control method which can shorten data forwarding time amount.

[0017]

[Means for Solving the Problem] Drawing 1 is the 1st principle block diagram of this invention, and drawing 2 is the 2nd principle block diagram of this invention.

[0018] In drawing 1 1 polygonal top-most-vertices data The included polygon data storage section and 2 a polygon the 1st triangle data with which the polygon division processing section divided into two or more triangles and 3 were stored in a triangle and the periphery line data storage section, and 4 was stored in a triangle and the periphery line data storage section 3, and the 2nd triangle -- the triangle data of data etc., and 4a-4c -- three square shapes each -- receiving -- the 1st -- or The side corresponding to the 3rd each top-most-vertices coordinate and each top-most vertices A boundary line (A polygonal periphery line) and a parting line (a polygon) When it divides, each top-most-vertices data constituted from a discernment flag which indicates whether to be the produced line, the data-processing section in which the drawing processing section and 5a process triangle data in 5, the attribute data storage section in which the triangle drawing section and 5c include the periphery line drawing section in, and, as for 6, 5b includes drawing conditions, and 7 are frame memories.

[0019] Moreover, as for the graphic form data storage section and 16, in drawing 2 , the periphery line linearity control section by which the polygon division processing section and 12 were prepared in the clip processing section, and 13 was prepared [10] for the polygon data storage section and 11 in the clip processing section 12, the periphery line connection processing section by which 14 was similarly prepared in the clip processing section 12, and 15 are [the drawing processing section and 17] frame memories. Each part of 10, 11, 16, and 17 of this drawing 2 corresponds to 1, 2, 5, and 7 of above-mentioned drawing 1 , respectively.

[0020] When dividing a polygon, the 1st configuration of this invention adds the flag which shows each triangular top-most-vertices data whether the side connected to the next top-most vertices is a boundary line, or it is a parting line, and can be made to carry out drawing processing of the periphery line at a high speed in the drawing processing section. Moreover, with the 2nd configuration, in the clip processing by the clip frame, triangulation is performed to the divided triangle, the flag showing whether the periphery line corresponding to each top-most vertices is drawn is added into the triangle data, the type of periphery drawing drawing is identified using the flag, and only the data of a required periphery

line are sent out to the drawing processing section.
[0021]
[Function] In the configuration of drawing 1 , the polygon division processing section 2 inputs the polygon data of the polygon data storage section 1, and divides a polygon into two or more triangles. The discernment flag which shows a boundary line and a parting line to the top-most vertices of three square shapes each created by this division is added, and it stores as triangle data 4 of the triangle data storage section 3. The triangle data 4 consist of three top-most-vertices data 4a-4c, and consist of a top-most-vertices coordinate and a discernment flag.
[0022] The triangle data 4 are transmitted to the drawing processing section 5, each top-most-vertices data is analyzed by data-processing section 5a in this, a triangle is drawn by the frame memory 7 by triangle drawing section 5b, and only the top-most vertices where the flag which expresses a boundary line with periphery line drawing section 5c with reference to the drawing conditions (periphery line visible and invisibility) of the attribute data storage section 6 when visible was attached draw a periphery line to a frame memory 7. The drawing processing section 5 draws similarly about all triangles.
[0023] In the configuration of drawing 2 , the polygon division processing section 11 inputs the polygon data of the polygon data storage section 10 like above-mentioned drawing 1 , and divides them into two or more triangles. or [that the periphery line corresponding to each triangular top-most-vertices coordinate and each top-most vertices which were divided by the clip processing section's 12 performing clip processing by the clip frame, and carrying out triangulation of the graphic form included in a frame to three divided square shape each data is visible] -- the triangle data with which the periphery line flag showing whether it is invisible was added are created. The periphery line flag corresponding to each top-most vertices is inputted into the periphery line linearity control section 13, and the Rhine flag showing a triangular drawing type (pattern of which side in three sides to draw) is created using a periphery line flag. Next, the periphery line connection processing section 14 connects top-most-vertices data only about the periphery line segment which should draw based on said Rhine flag, and stores in the graphic form data storage section 15. The contents of this graphic form data storage section 15 are sent to the drawing processing section 16. The drawing processing section 16 performs processing of triangle drawing and periphery drawing drawing, and stores it in a frame memory 17.
[0024]
[Example] Drawing 3 is the example of drawing of five square shapes according [drawing 5] to an example 1 according [the block diagram of an example 1 and drawing 4] to the example of a configuration of triangle data.
[0025] This example 1 is a configuration corresponding to the 1st principle configuration of this invention shown in above-mentioned drawing 1 . In drawing 3 , 2-4, and 5 and 7 are equipment equipped with the same name as above-mentioned drawing 1 , and the detail inside the drawing processing section 5 is shown especially. [0026] If actuation of an example 1 is explained below, the polygon division processing section 2 receives the polygon data (101 of this drawing 12) from the polygon data storage section (it is the same as 100 of drawing 12 of the conventional example) which is not illustrated, performs processing which divides this into two or more triangles, and stores the divided data of three square shapes each in a triangle and the periphery line data storage section 3. this time -- three square shapes each -- the 1st triangle data and the 2nd triangle data -- sequential creation is carried out with .. For example, when 5 square shape data input as polygon data, the 1st, 2nd, and 3rd triangle data is created.
[0027] As three square shape each data, the polygon division processing section 2 of this invention sets the discernment flag showing any of a boundary line or a parting line the periphery lines corresponding to the top-most vertices of opposite Perilla frutescens (L.) Britton var. crispa (Thunb.) Decne. are as each top-most-vertices coordinate while generating the 1st top-most-vertices coordinate thru/or the 3rd top-most-vertices coordinate. If this discernment flag is the periphery line which "1" was set up as a boundary layer flag and produced by division when the periphery line corresponding to each top-most vertices expresses the boundary of the original polygon, "0" will be set up as a parting line flag.

[0028] In the example of a configuration of the triangle data shown in drawing 4 , as one top-most-vertices coordinate, each coordinate data (32 bits each) of X, Y, and Z is set as 40-42, and the flag data 43 are set as this. The flag data 43 consist of 31 bits [0 to] a total of 32 bits, and "1" (boundary layer flag) and "0" (parting line flag) are set as 16 bits of them.
[0029] Thus, in a triangle and the periphery line data storage section 3, with each top-most-vertices coordinate and a discernment flag, the data 4a-4c of three square shapes each created by the polygon division processing section 2 are stored, as shown in drawing.
[0030] A triangle and the periphery line data storage section 3 transmit one triangle to a unit, and transmits triangle data to the drawing processing section 5. In the drawing processing section 5, triangle data are inputted in a triangle and the periphery line data input section 50, and processing is performed by the data-processing section 51 (it corresponds to 5a of drawing 1). first, in the n-th triangle data analysis (the number of the arbitration to the number of triangular with which n was created by division at the maximum), each top-most-vertices data of the n-th triangle is analyzed (S1 in 51 of drawing 3), and a triangle and periphery drawing drawing data are created based on an analysis result as the flow of processing of the data-processing section 51 is shown in the data-processing section 51 of drawing 3 (said -- S2). In this case, the discernment flag of a triangular top-most-vertices coordinate and a corresponding periphery line is united, and is created. whether while the created data are sent to the drawing section 52, they have other triangle data, and when distinguish (said -- S3) and it is, processing same about the following triangle and periphery line data is performed.
[0031] About the sent triangle and periphery drawing drawing data, the drawing section 52 performs the n-th triangle drawing in the triangle drawing section 53 (it corresponds to 5b of drawing 1). The 1st triangle is drawn to a frame memory 7 at first, and the periphery line of the 1st triangle is continuously drawn in the periphery line drawing section 54. the drawing conditions (periphery line visible / invisibility) stored in the attribute data storage section 6 at this time -- referring to -- a periphery line -- it is visible and only the periphery line corresponding to the top-most vertices where the boundary layer flag was added is drawn to a frame memory 7.
[0032] If the triangle and periphery drawing drawing data of the 2nd following triangle are sent from the data-processing section 51, the triangle and periphery line will be drawn similarly and it will be drawn on a frame memory 7 about all triangle and periphery drawing drawing data.
[0033] If the example of drawing 5 explains, when dividing into three triangles five square shapes shown in A. and drawing, the 1st triangle Each top-most-vertices coordinate of top-most vertices 1, 2, and 3, A boundary layer flag is added, a boundary layer flag is added also for the periphery line (line which connects top-most vertices 2 and top-most vertices 3) corresponding to top-most vertices 2, and, as for the periphery line (line which connects top-most vertices 1 and top-most vertices 2) corresponding to top-most vertices 1, the parting line flag is added, as for the periphery line (line which connects top-most vertices 3 and top-most vertices 1) corresponding to top-most vertices 3.
[0034] As a result of processing the data of such 1st triangle in the data-processing section 51 and drawing in the drawing section 52, two periphery lines (between between top-most vertices 1 and 2 and top-most vertices 2, and 3) shown by the 1st triangle (shading display) constituted from top-most vertices 1, 2, and 3 as shown in B. of drawing 5 , and the thick wire are drawn together. Then, it is drawn as drawing of the 2nd triangle equipped with the data showing a periphery line (between top-most vertices 3 and 4) with an one boundary layer flag of top-most vertices 3 shows C. of drawing 5 , and finally the 3rd triangle is drawn like D. of drawing 5 , and drawing of the whole 5 square shape is completed.
[0035] Next, drawing 6 thru/or drawing 10 are drawings about the configuration of the example 2 corresponding to the 2nd principle configuration of this invention shown in above-mentioned drawing 2 . In the configuration shown in drawing 2 , clip processing is performed in quest of the intersection of a clip frame and a triangle to one triangle divided in the polygon division processing section 11 as usual in the clip processing section 12. Drawing 6 explains this example.
[0036] Drawing 6 is the example of the triangle clipped by clip processing, the divided triangle consists of top-most vertices A, B, and C, and this triangle and the clip frame D cross by point B' and point C'.

Therefore, top-most vertices A, B', and the triangle constituted from C' are newly divided by clip processing from the triangle constituted from original top-most vertices A, B, and C.
[0037] each periphery line of the triangle (triangle of the top-most vertices A of drawing 6 , B', and C') divided with a clip in the clip processing section 12 -- drawing (visible) and drawing -- being nothing (invisibility) -- the periphery line flag with which it expresses is created. namely, -- since both the segments L1 (periphery line between top-most vertices A and top-most-vertices B') and segments L2 (periphery line between top-most vertices A and top-most-vertices C') of drawing 6 are segments on the periphery line of the original triangle -- being visible (drawing), although a periphery line flag is set as the condition of expressing top-most-vertices B' and top-most-vertices C' -- the segment of a between has a periphery line flag set as the condition of not drawing since it is the periphery line produced in division by clip processing
[0038] Drawing 7 is the explanatory view of a periphery line flag, and, as for the block diagram of a periphery line flag, and B., A. of drawing 7 shows the relation between top-most vertices and a periphery line flag. the time of a periphery line flag being set as the location of 0 bit in 32-bit data, as shown in A., and this 0 bit being "1" -- a periphery line -- visible (those with drawing), and when it is "0", periphery line invisibility (with no drawing) is expressed. As shown in B., the periphery line flag of the periphery line (en: mean Edge n) of top-most vertices n and top-most vertices n+1 is added to the coordinate data of top-most vertices n, and this periphery line flag is en+1 (the periphery line flag of an edge n+1 is added en+2 on top-most vertices n+2) to the coordinate data of top-most vertices n+1 similarly. The periphery line flag of (an edge n+2) is added.
[0039] Drawing 8 and drawing 9 are the processing flow Fig. (the 1) of an example 2, and (its 2). This processing flow shows the function of the periphery line linearity control section (13 of drawing 2) which operates by control of the clip processing section (12 of drawing 2), and the periphery line connection processing section (14 of drawing 2).
[0040] Moreover, the triangle equipped with the top-most vertices P1, P2, and P3 which show this processing flow to the right-hand side of drawing 8 by the clip processing section shall be created, the periphery line flag of each edge shall be added to the coordinate data of each top-most vertices by the same relation as the above and B. of drawing 7 , and the periphery line flag of each edge is set as any of "0" or "1", or arbitration.
[0041] Rhine flag (LINEFLG and display) which processing of a periphery line linearity control section was performed, and was formed in the specific area of internal memory (not shown) when processing of drawing 8 began ** -- the variable (specific area of memory) of the identifier to say is initialized to 0 (S1 of drawing 8). Next, the following processings are performed in step S2. That is, it is the Rhine flag (LINEFLG) to the beginning. It receives and the contents of the periphery line flag of an edge 1 (edge1) are set up. At this time, the contents (1 or 0) which express the flag set as 0 bit positions in 32 bits as shown in A. of above-mentioned drawing 7 are the Rhine flags (LINEFLG). It is set as 0 bit.
[0042] Next, 1 bit shift of the periphery line flag of an edge 2 (edge2) is carried out to left-hand side, and it is the Rhine flag (LINEFLG) about the bit (since 1 bit shift is carried out to the left, it is in 1 bit position in 32 bits) of the periphery line flag. It is set as corresponding 1 bit position. Furthermore, 2 bit shifts of the periphery line flag of an edge 3 (edge3) are carried out to the left, and it is the Rhine flag (LINEFLG) about the bit (it is in 2 bit positions in 32 bits) of the Rhine flag. It sets up.
[0043] The explanatory view of the Rhine flag shown in drawing 10 explains the contents of processing of the above-mentioned step S2. In A. of drawing 10 , ** - ** show the condition after performing the left shift to whom the periphery line flag of an edge 1 - an edge 3 corresponds, and ** expresses the Rhine flag. ** As the contents of each 0 bit position (flag bit) which is -** show **, it is set as the three bit positions of the Rhine flag.
[0044] Rhine flag (LINEFLG) which returned to drawing 8 and then was obtained ******** -- the contents judge whether it is 0 (S3 of drawing 8). It is the Rhine flag (LINEFLG) by this decision. When all the contents are 0 (all bits are "0"), it ends without carrying out periphery line connection processing, since there is no periphery line which draws. the case where the contents are not 0 -- Rhine flag (LINEFLG) A jump place is set up (S4 of drawing 8) and jumped at the set-up JAMBU point (any of

M1-M7: mention later about M1-M7) (said -- S5). This jump place is the periphery line connection processing section shown in drawing 9 .
[0045] Here, it is the above-mentioned Rhine flag (LINEFLG). When drawing 10 explains an operation, it is the Rhine flag (LINEFLG) by A. of drawing 10 . The periphery line flag of an edge 3 is set as the periphery line flag of an edge 1 by the triplet in the lowest 0 bit positions, and is set as 1 bit position in the periphery line flag of an edge 2, and 2 bit positions, respectively. This Rhine flag (LINEFLG) The range expressed with a triplet is 000-111, and becomes eight values of 0-7 of a decimal number.
[0046] The table of B. of drawing 10 is this Rhine flag (LINEFLG). Eight values show that the pattern of drawing for a periphery drawing streak is expressed, respectively, and show the example of a pattern for a periphery drawing streak corresponding to this to drawing 11 .
[0047] namely, B. of drawing 10 -- setting -- number (1) ** the case where the Rhine flag is 0 -- it is -- drawing -- being nothing -- a jump place -- M0 (when judged as 0 at step S3 of drawing 8) -- becoming -- a part for a periphery drawing streak -- (1) of drawing 11 It is not transmitted so that it may be shown. moreover, number (2) ** It is the case where the Rhine flag is 1, and it directs that the amount of periphery drawing streak draws between Ptop-most-vertices P1-2 of the triangle shown in the right-hand side of drawing 8 , and the periphery line edge to be used is an edge 1, and a jump place is set to M1. In this case, it is (2) of drawing 11 at a jump place. Shown drawing is performed.
[0048] Similarly, it is a number (3). - (8) Rhine flag (LINEFLG) "2" Directing that each top-most-vertices section shown in a part for each periphery drawing streak of B. of drawing 10 draws using a corresponding periphery line edge corresponding to the value of - "7", each jump place is the Rhine flag (LINEFLG). It is M2-M7 corresponding to a numeric value.
[0049] The processing flow of the periphery line connection processing section jumped from the periphery line linearity control section of above-mentioned drawing 8 is shown in drawing 9 , periphery line connection processing in which respond for any of M1-M7 which were set up as a jump place being, and it corresponds is performed, and return termination is carried out at drawing 8 after activation. In addition, M1-M7 are each number (2) of B. of above-mentioned drawing 10 . - (7) It corresponds to a case.
[0050] For example, when a jump place is M1, the data of the 1st top-most vertices (P1 of the triangle of drawing 8) are stored in the graphic form data storage section (15 of drawing 2), and then the data of the 2nd top-most vertices (P2 of the triangle of drawing 8) are stored. Thereby, it is (2) of drawing 11 . A periphery line as shown is drawn.
[0051] moreover It corresponds, when drawing two periphery lines of the triangle created with the clip shown in above-mentioned drawing 6 , and drawing between Ptop-most-vertices P2-P3-1 of the triangle of drawing 8 . At this time, it is the Rhine flag (LINEFLG). A triplet is set as "110", and since it is "6" of a decimal number, it is jumped to M6. In this case, if storing (to graphic form data storage section) of the data of the 2nd top-most vertices (P2), storing of the data of the 3rd top-most vertices, and storing of the data of the 1st top-most vertices are performed one by one and three top-most-vertices data are stored by the flow of drawing 9 , these data will be drawn by the frame memory in the drawing processing section (16 of drawing 2).
[0052] In this case, two periphery line segments are connected, and it is three pieces, and the top-most-vertices data transmitted to the drawing processing section do not need to transmit two top-most-vertices data of a starting point-terminal point for every segment like the conventional example, can lessen memory space, and can shorten the transfer time.
[0053]
[Effect of the Invention] Since according to the 1st principle of this invention it can draw by adding the discernment flag showing a boundary line and a parting line to top-most-vertices data, and processing this in case a polygon is divided into two or more triangles, memory can be contraction-ized while accelerating periphery line processing. That is, creation processing of periphery line data can be simplified, the amount of data transfer between the polygon division processing section and the drawing processing section can be reduced, and periphery line data processing in drawing processing can be reduced.

[0054] When drawing a triangular periphery line according to the 2nd principle of this invention, after the triangle which carried out polygon division clips, when managing the periphery line top-most-vertices data of a complicated triangle, it becomes possible to use a periphery line flag and to simplify data. Moreover, periphery line top-most-vertices data required for drawing can be reduced, and data forwarding time amount can be shortened.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

MEANS

[Means for Solving the Problem] Drawing 1 is the 1st principle block diagram of this invention, and drawing 2 is the 2nd principle block diagram of this invention.
[0018] In drawing 1 1 polygonal top-most-vertices data The included polygon data storage section and 2 a polygon the 1st triangle data with which the polygon division processing section divided into two or more triangles and 3 were stored in a triangle and the periphery line data storage section, and 4 was stored in a triangle and the periphery line data storage section 3, and the 2nd triangle -- the triangle data of data etc., and 4a-4c -- three square shapes each -- receiving -- the 1st -- or The side corresponding to the 3rd each top-most-vertices coordinate and each top-most vertices A boundary line (A polygonal periphery line) and a parting line (a polygon) When it divides, each top-most-vertices data constituted from a discernment flag which indicates whether to be the produced line, the data-processing section in which the drawing processing section and 5a process triangle data in 5, the attribute data storage section in which the triangle drawing section and 5c include the periphery line drawing section in, and, as for 6, 5b includes drawing conditions, and 7 are frame memories.
[0019] Moreover, as for the graphic form data storage section and 16, in drawing 2 , the periphery line linearity control section by which the polygon division processing section and 12 were prepared in the clip processing section, and 13 was prepared [10] for the polygon data storage section and 11 in the clip processing section 12, the periphery line connection processing section by which 14 was similarly prepared in the clip processing section 12, and 15 are [the drawing processing section and 17] frame memories. Each part of 10, 11, 16, and 17 of this drawing 2 corresponds to 1, 2, 5, and 7 of above-mentioned drawing 1 , respectively.
[0020] When dividing a polygon, the 1st configuration of this invention adds the flag which shows each triangular top-most-vertices data whether the side connected to the next top-most vertices is a boundary line, or it is a parting line, and can be made to carry out drawing processing of the periphery line at a high speed in the drawing processing section. Moreover, with the 2nd configuration, in the clip processing by the clip frame, triangulation is performed to the divided triangle, the flag showing whether the periphery line corresponding to each top-most vertices is drawn is added into the triangle data, the type of periphery drawing drawing is identified using the flag, and only the data of a required periphery line are sent out to the drawing processing section.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

OPERATION

[Function] In the configuration of drawing 1 , the polygon division processing section 2 inputs the polygon data of the polygon data storage section 1, and divides a polygon into two or more triangles. The discernment flag which shows a boundary line and a parting line to the top-most vertices of three square shapes each created by this division is added, and it stores as triangle data 4 of the triangle data storage section 3. The triangle data 4 consist of three top-most-vertices data 4a-4c, and consist of a top-most-vertices coordinate and a discernment flag.
[0022] The triangle data 4 are transmitted to the drawing processing section 5, each top-most-vertices data is analyzed by data-processing section 5a in this, a triangle is drawn by the frame memory 7 by triangle drawing section 5b, and only the top-most vertices where the flag which expresses a boundary line with periphery line drawing section 5c with reference to the drawing conditions (periphery line visible and invisibility) of the attribute data storage section 6 when visible was attached draw a periphery line to a frame memory 7. The drawing processing section 5 draws similarly about all triangles.
[0023] In the configuration of drawing 2 , the polygon division processing section 11 inputs the polygon data of the polygon data storage section 10 like above-mentioned drawing 1 , and divides them into two or more triangles. or [that the periphery line corresponding to each triangular top-most-vertices coordinate and each top-most vertices which were divided by the clip processing section's 12 performing clip processing by the clip frame, and carrying out triangulation of the graphic form included in a frame to three divided square shape each data is visible] -- the triangle data with which the periphery line flag showing whether it is invisible was added are created. The periphery line flag corresponding to each top-most vertices is inputted into the periphery line linearity control section 13, and the Rhine flag showing a triangular drawing type (pattern of which side in three sides to draw) is created using a periphery line flag. Next, the periphery line connection processing section 14 connects top-most-vertices data only about the periphery line segment which should draw based on said Rhine flag, and stores in the graphic form data storage section 15. The contents of this graphic form data storage section 15 are sent to the drawing processing section 16. The drawing processing section 16 performs processing of triangle drawing and periphery drawing drawing, and stores it in a frame memory 17.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] It is the 1st principle block diagram of this invention.
[Drawing 2] It is the 2nd principle block diagram of this invention.
[Drawing 3] It is the block diagram of an example 1.
[Drawing 4] It is drawing showing the example of a configuration of triangle data.
[Drawing 5] It is drawing showing the example of drawing of five square shapes by the example 1.
[Drawing 6] It is drawing showing the example of the triangle clipped by clip processing.
[Drawing 7] It is the explanatory view of a periphery line flag.
[Drawing 8] It is the processing flow Fig. (the 1) of an example 2.
[Drawing 9] It is the processing flow Fig. (the 2) of an example 2.
[Drawing 10] It is the explanatory view of the Rhine flag.
[Drawing 11] It is drawing showing the example of a pattern for a periphery drawing streak.
[Drawing 12] It is the block diagram of the conventional periphery drawing drawing.
[Drawing 13] It is drawing showing the example of the periphery drawing drawing by the conventional example of operation.
[Drawing 14] It is the explanatory view of the control system of the triangular periphery line data divided by the conventional clip processing.
[Description of Notations]
1 Polygon Data Storage Section
2 Polygon Division Processing Section
3 Triangle and Periphery Line Data Storage Section
4 Triangle Data
4a-4c Each top-most-vertices data
5 Drawing Processing Section
5a Data-processing section
5b Triangle drawing section
5c Periphery line drawing section
6 Attribute Data Storage Section
7 Frame Memory

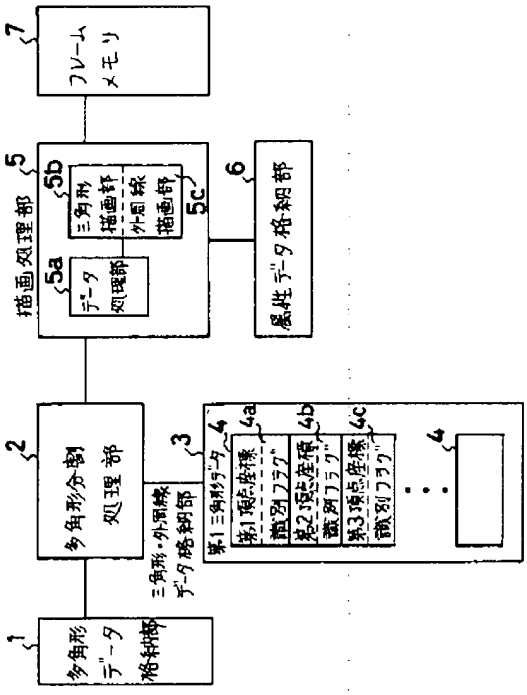
[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

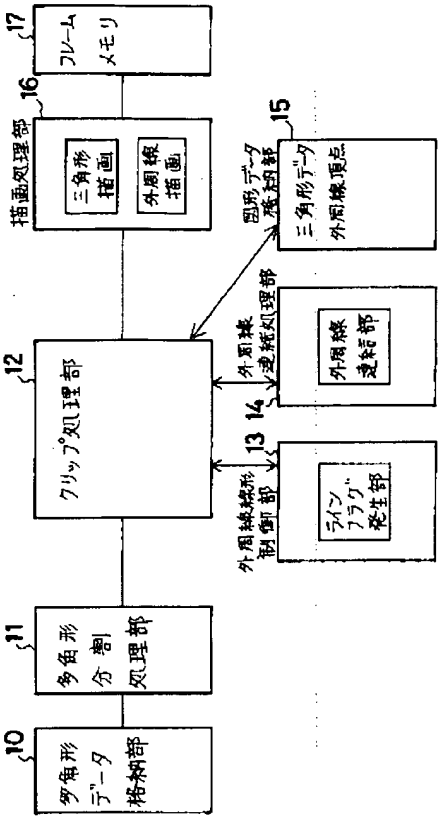
DRAWINGS

[Drawing 1]
本発明の第1の原理構成図



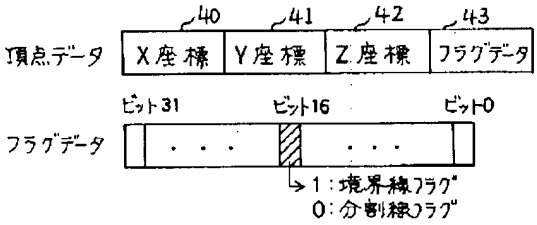
[Drawing 2]

本発明の第2の原理構成図



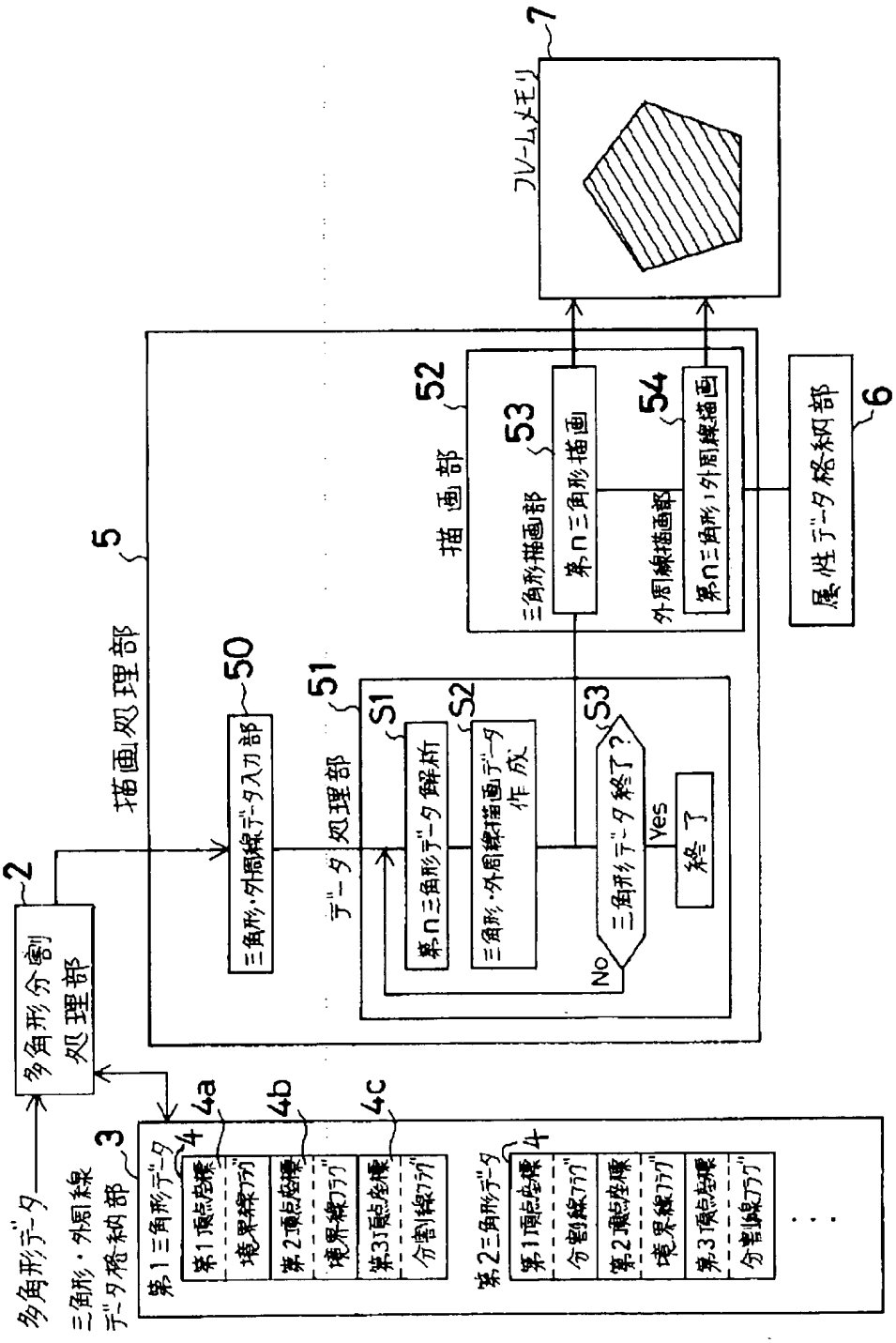
[Drawing 4]

三角形データの構成例



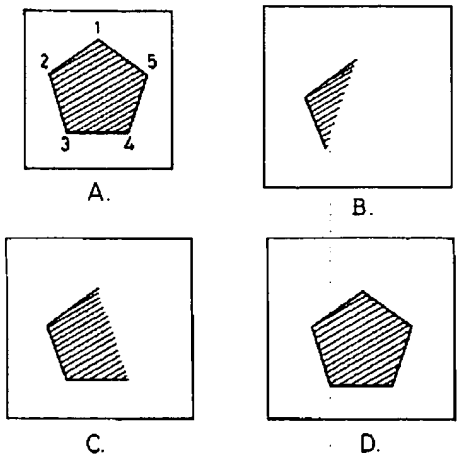
[Drawing 3]

実施例1の構成図

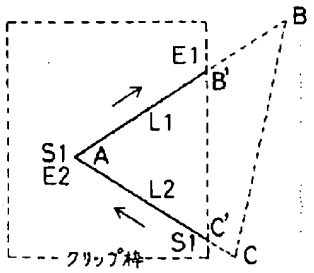


[Drawing 5]

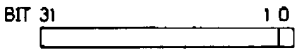
実施例1による5角形の描画の例



[Drawing 6]
クリップ処理によりクリップされる三角形の例

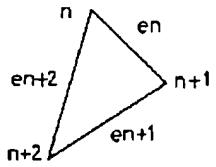


[Drawing 7]
外周線フラグの説明図



0ビット 1：外周線可視(描画有り)
0：外周線不可視(描画無し)

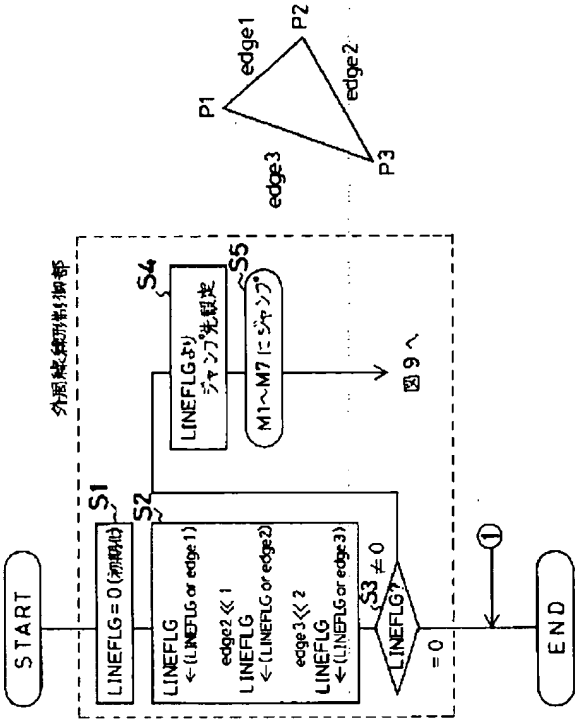
A. 外周線フラグの構成図



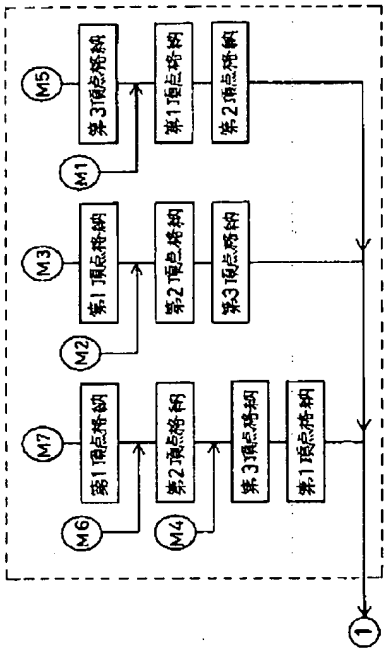
B. 頂点と外周線フラグの関係図

[Drawing 8]

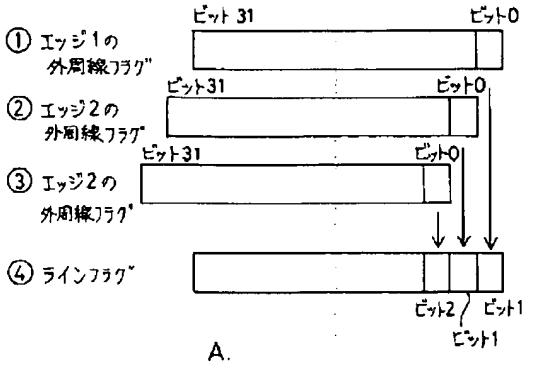
実施例２の処理フロー図（その１）



[Drawing 9]
実施例２の処理フロー図（その２）



[Drawing 10]
ラインフラグの説明図



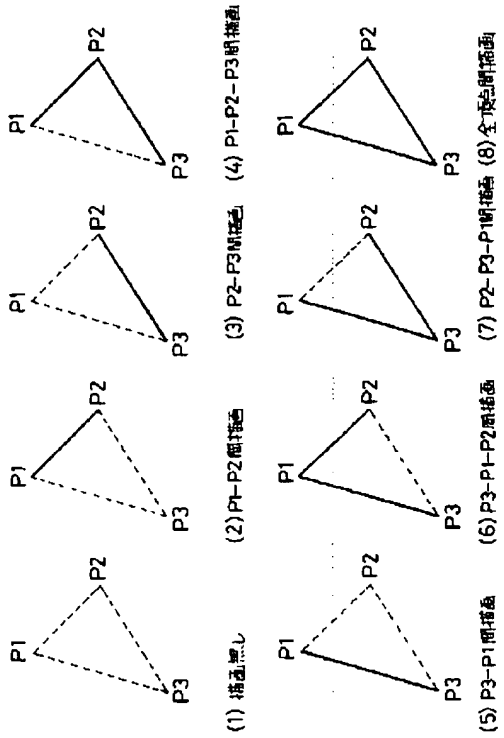
A.

	外周線描画線分	使用する外周線エッジ	LINEFLG	ジャンプ先
(1)	描画無し	外周線エッジなし	0	M0
(2)	P1-P2間描画	edge1	1	M1
(3)	P2-P3間描画	edge2	2	M2
(4)	P1-P2-P3間描画	edge1,edge2	3	M3
(5)	P3-P1間描画	edge3	4	M4
(6)	P3-P1-P2間描画	edge3,edge1	5	M5
(7)	P2-P3-P1間描画	edge2,edge3	6	M6
(8)	全頂点間描画	edge1,edge2,edge3	7	M7

B.

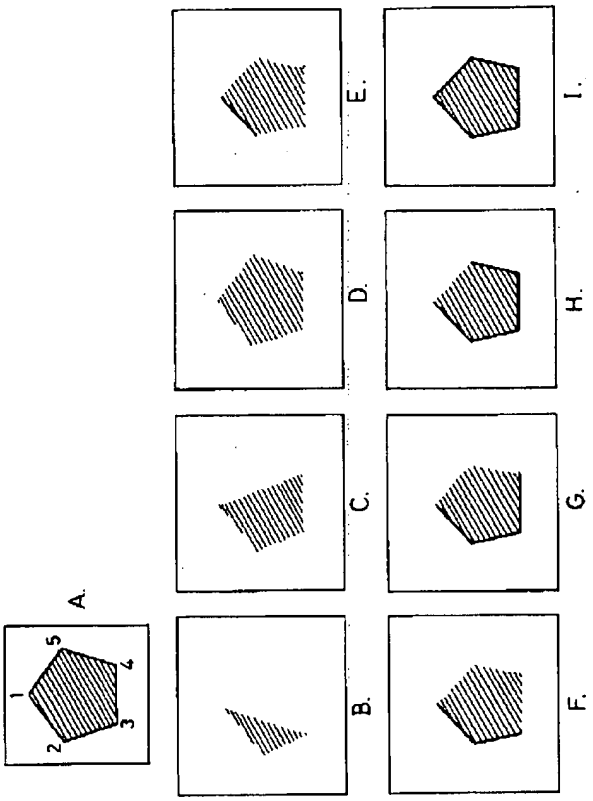
[Drawing 11]

外周線描画線分のパターン例を示す図



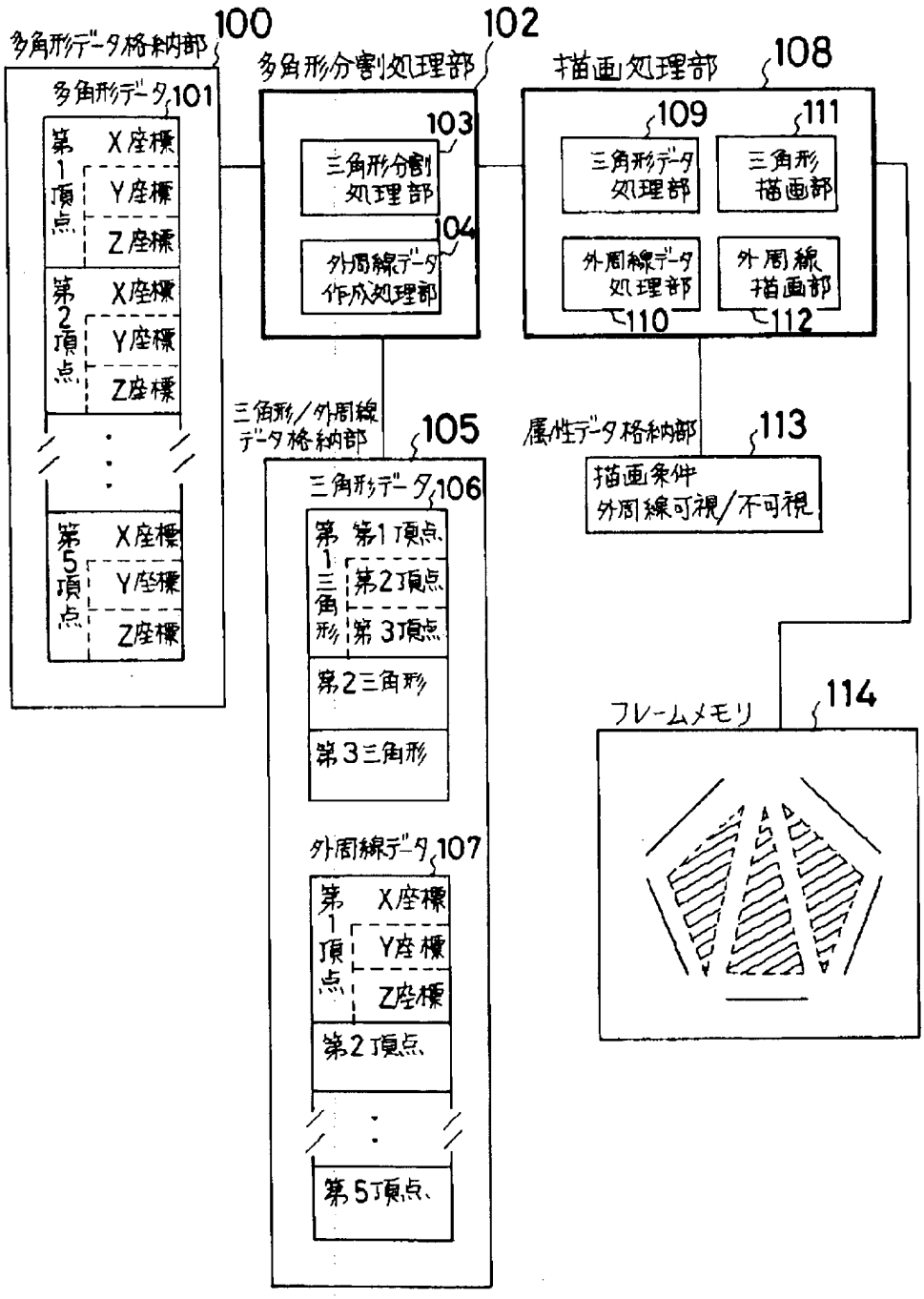
[Drawing 13]

従来例による外周線描画の動作例を示す図



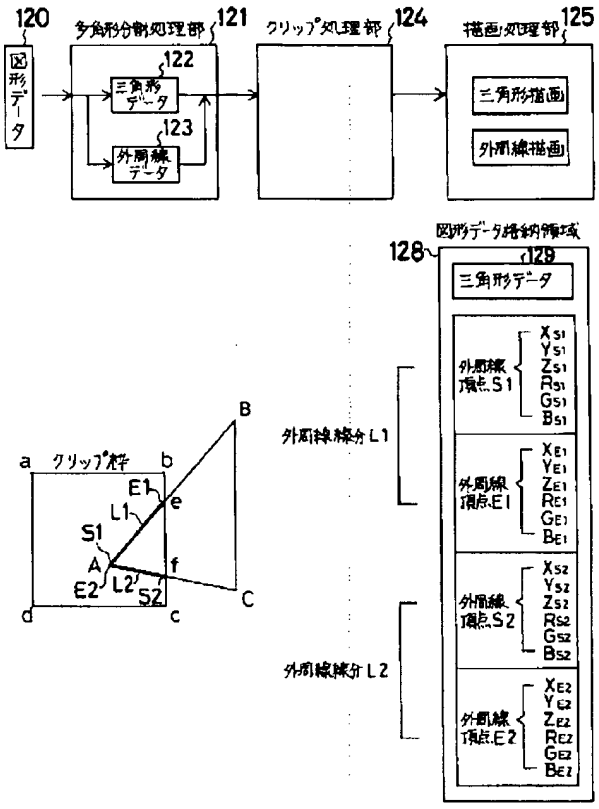
[Drawing 12]

従来の外周線描画の構成図



[Drawing 14]

従来のクリップ処理で分割した三角形の外周線データの
制御方式の説明図



[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] The polygon division processing section which inputs polygon data and is divided into two or more triangles in the periphery drawing drawing data control method which draws by dividing a polygon into two or more triangles Create the triangle data constituted from a discernment flag showing any of a boundary line or a parting line the periphery lines corresponding to each top-most-vertices coordinate and each top-most-vertices coordinate of three divided square shapes each are, and the drawing processing section said three divided square shape each data which were created in said polygon division processing section The periphery drawing drawing data control method characterized by receiving and drawing a polygon by processing the periphery line of three square shapes each based on said discernment flag.
[Claim 2] Said drawing processing section is a periphery drawing drawing data control method characterized by drawing only the periphery line corresponding to the top-most vertices where drawing and said boundary layer flag of said three square shape each data were added in claim 1.
[Claim 3] The periphery drawing drawing data control method characterized by drawing only the periphery line by which periphery line visible ****** is carried out with reference to the drawing conditions beforehand set as the attribute data storage section in claim 1 when drawing said periphery line.
[Claim 4] In the periphery drawing drawing data control method which draws after dividing the polygon into two or more triangles and starting by clip processing As opposed to the triangle data which the polygon division processing section which inputs polygon data and is divided into two or more triangles created three divided square shape each data, and created the clip processing section which performs clip processing in said polygon division processing section with a clip frame Triangulation is performed, triangle data equipped with the periphery line flag showing whether each triangular periphery line is drawn to the triangle generated with a clip are generated, and said each periphery line flag is used. With a clip The periphery drawing drawing data control method characterized by connecting the top-most-vertices data of three square shapes each using said each periphery line flag when the generated triangle data are sent out to the drawing processing section.
[Claim 5] In claim 4, said clip processing section by whether it is the periphery line of the original triangle before clip processing is carried out Set up said periphery line flag, add a periphery line flag for each [to which the triangle divided with a clip corresponds] top-most-vertices data of every, and the periphery line flag for every top-most vertices of said triangle is processed. The periphery drawing drawing data control method characterized by having the periphery line linearity control section which generates the Rhine flag constituted from two or more bits showing the pattern of the periphery line drawn.
[Claim 6] It is the periphery drawing drawing data control method characterized by having the periphery line connection section which said clip processing section connects the top-most-vertices data of a periphery segment in claim 5 based on the pattern of said Rhine flag for drawing processing, and is stored in the graphic form data storage section.

[Translation done.]